

Universidad CARLOS III de Madrid

Diseño y fase de testing de un videojuego en Blender.

Ingeniería de Telecomunicaciones Sonido e Imagen



AUTOR: Ana Román Rodríguez
TUTOR: Juan Peralta Donante

Miércoles, 31 de agosto de 2011

AGRADECIMIENTOS

A la primera persona a la que quiero agradecer este proyecto es a mi tutor Juan, por confiar en mí a pesar de no cursar ingeniería informática y por darme ánimos en los momentos en que flaqueaba y pensaba que no iba a ser capaz de sacar este proyecto adelante.

A mi familia, mi padre Rafael y mi madre Encarna porque, a pesar de las circunstancias, habéis insistido día a día en darme una buena educación y formación, sin poner pega alguna a mis decisiones y dándome mucha fuerza en los momentos más duros fuera y dentro de la carrera. A mi hermano “Rafita”, por apoyarme en los malos momentos y por darme mucha alegría en los buenos, porque aunque, a veces tengamos nuestros más y nuestros menos, espero que haya sido y sea un buen ejemplo para ti. Y al resto de mi familia, porque son los mejores.

A mis amigos “los de Boadilla”. A los chicos, Jose, Jaime, Álvaro, Jorge, Alex, Guti, Pablo, Elias y Willy, por haber pasado grandes momentos durante y después de la adolescencia y porque no decirlo, grandes fiestas. A mis chicas, mis “7 pecados”: María, Clara, Martita, Ri, Elsa, Cynthia, y Vicky, por haber sido mis confidentes, mi paño de lágrimas, mis mejores tardes de parque, mis mejores vacaciones, mis mejores años y mi vida. Con mención especial para Jaime, mi filósofo favorito, y a Vicky, simplemente porque te quiero.

A mis amigos “los de la UNI”, Ana Belén, Javi, Cris, David (tu eres uno más), Marta, Guille, Laura, Carlos, Alvarito y Juanlu, porque sin vosotros la universidad no habría sido lo mismo, esas tardes de prácticas interminables, esas horas en la biblioteca con el cerebro exprimido, esas risas en la cafe, tantos y tantos momentos que nunca, y repito, nunca olvidaré. A los “sonimágicos” por grandes momentos.

Y por último, a vosotras, Mery, Olaia, Dani y Ana, por esas tardes en “La Buhardilla”, por esos muses interminables, por esos magníficos entrenamientos de vóley y por haberos encontrado, porque sois muy importantes para mí.

ÍNDICE

1. INTRODUCCIÓN	9–11
1.1. OBJETIVOS DEL PROYECTO	11
2. ESTADO DE LA CUESTIÓN.....	13–47
2.1. MODELADO 3D.....	13–23
2.1.1. Cronológicamente	14–23
2.1.2. Herramienta de modelado 3D	23–28
2.2. MOTORES DE JUEGO	28–44
2.2.1. Cronológicamente	29–40
2.2.2. Motores de juego en la actualidad	42–44
2.3. BLENDER ENGINE	45–47
3. ANÁLISIS	49–64
3.1. PROPUESTA INICIAL.....	49–50
3.2. ANÁLISIS DE LAS HERRAMIENTAS A UTILIZAR.....	50–52
3.3. ANÁLISIS DE LOS REQUISITOS.....	52–56
3.4. CASO DE USO	57
3.5. DIAGRAMA DE EJECUCIÓN	58
3.6. ESTIMACIÓN Y PRESUPUESTO INICIAL.....	63
4. DESARROLLO DEL PROYECTO.....	64–103

4.1. INFLUENCIAS	64–65
4.2. CONCEPCIÓN	66
4.3. DISEÑO	67–98
4.3.1. Historia.....	67
4.3.2. Arte conceptual	67–69
4.3.3. Sonido	69–70
4.3.4. Mecánica del juego	70–75
4.3.5. Diseño de las plataformas	75–82
4.3.6. Diseño de la programación	82–86
4.3.7. Ejemplos de la implementación	86–98
4.4. POST-PRODUCCIÓN.....	98–103
4.4.1. Efectos de sonido.....	99
4.4.2. Interfaz gráfica de usuario	99–103
4.4.3. Generación del archivo ejecutable	103
5. FASE DE TESTING	104–37
5.1. LA JUGABILIDAD	104
5.2. JUGABILIDAD vs USABILIDAD.....	104–5
5.3. PROPIEDADES DE LA JUGABILIDAD.....	105–12
5.3.1. Satisfacción	105–6
5.3.2. Aprendizaje	106–8
5.3.3. Efectividad	108–9
5.3.4. Inmersión	109–10
5.3.5. Motivación	110
5.3.6. Emoción	110–11
5.3.7. Socialización.....	111–12
5.4. FACETAS DE LA JUGABILIDAD	112–13

5.5. MI PROPIA FASE DE TESTING	113–35
5.5.1. Test de pruebas	114–16
5.5.2. Test de jugabilidad.....	116–35
5.6. CONCLUSIÓN DE LA FASE DE TESTING.....	135
6. CONCLUSIONES Y LÍNEAS FUTURAS	137–45
6.1. SOBRE LA REALIZACIÓN DEL PROYECTO.	137
6.2. PLANIFICACIÓN DEL PROYECTO.....	140
6.2.1. Búsqueda de las posibles tecnologías	138
6.2.2. Iniciación con la herramienta Blender.....	138
6.2.3. Desarrollo del modelado	139
6.2.4. Desarrollo de la implementación	139
6.2.5. Perfeccionamiento.....	139
6.2.6. Fase de testing.....	139
6.2.7. Redacción de la memoria	140
6.3. ESTIMACIÓN FINAL.....	142
6.4. PRESUPUESTO FINAL	145
6.5. LÍNEAS FUTURAS	145
7. BIBLIOGRAFÍA	147–48
8. REFERENCIAS	150
ANEXOS.....	152–88
ANEXO I. Acrónimos y definiciones	155–57
ANEXO II. Diagram de GANTT inicial.....	158

ANEXO III. Diagrama de PERT inicial	159–62
ANEXO IV. Diagram de GANTT final.....	163–65
ANEXO V. Diagrama de PERT final	165
ANEXO VI. Test de Jugabilidad.....	168
ANEXO VII. Test del grupo de menores de 30 años	179–88
ANEXO VIII. Test del grupo de mayores de 30 años.....	188

ÍNDICE DE CONTENIDOS

Ilustraciones

Ilustración 1. Cartel de la película FutureWorld	16
Ilustración 2. Cartel de la película WestWorld.....	16
Ilustración 3. Imagen de la película " Adam Power and the Jungle".	16
Ilustración 4. Imagen de la película "Tron".	17
Ilustración 5. Imagen "Terminator 2".	18
Ilustración 6. Imagen de la película "Titanic".	18
Ilustración 7. Imagen película "Jurassic Park".	19
Ilustración 8. Imagen del spot publicitario de Coca-Cola.	19
Ilustración 9. Cartel de la película "Toy Story"	20
Ilustración 10. Imagen de la película " ToyStory.	19
Ilustración 11. Imagen videojuego Quake.	20
Ilustración 12. Escena de la película "La Comunidad del Anillo"	21
Ilustración 13. Eje cronológico del los proyectos que incluyen animación por ordenador hasta la actualidad.	23
Ilustración 14. Ultima Underworld.	33
Ilustración 15. ZZT.	34
Ilustración 16. Wolf 3D.	34
Ilustración 17. Doom.	35
Ilustración 18. Quake I.	35
Ilustración 19. Counter Strike.	36
Ilustración 20. Unreal.	37
Ilustración 21. Grand Theft Auto.	37

Ilustración 22. Tirbes 2.	38
Ilustración 23. Doom III.	39
Ilustración 24. Left 4 Dead.	40
Ilustración 25. Grand Theft Auto IV.	40
Ilustración 26. Eje cronológico de los motores de juegos vistos hasta el momento.	42
Ilustración 27. Diagrama de ejecución.	59
Ilustración 28. Diagrama de GANTT inicial.	61
Ilustración 29. Diagrama de PERT inicial.	62
Ilustración 30. Carcasa del juego "Mercury" para PSP.	66
Ilustración 31. Pantalla "Mercury".	66
Ilustración 32. Pantalla "Mercury".	66
Ilustración 33. Story Board Tablero 1.	69
Ilustración 34. Story Board Tablero 2.	69
Ilustración 35. Story Board Tablero 3.	69
Ilustración 36. Story Board GUI.	70
Ilustración 37. Coordenadas de referencia.	72
Ilustración 38. Ventana por defecto.	77
Ilustración 39. Transformaciones básicas.	80
Ilustración 40. Ventanas de materiales y texturizado.	81
Ilustración 41. Nivel 1.	81
Ilustración 42. Nivel 2.	82
Ilustración 43. Nivel 3.	82
Ilustración 44. Diagrama de ciclo de vida.	83
Ilustración 45. Blender workspace ejemplo 1.	87
Ilustración 46. Ventana de implementación del objeto <i>CirculoAbrePuerta</i> .	88
Ilustración 47. Sensors, Controllers & Actuators.	89
Ilustración 48. Blender workspaces ejemplo 2.	89
Ilustración 49. Ventana de implementación del objeto <i>Sphere</i> .	90
Ilustración 50. <i>Sensors, Controllers & Actuators</i> .	91
Ilustración 51. Blender workspace ejemplo 3.	92
Ilustración 52. <i>Sensors, Controller & Actuator</i> .	93
Ilustración 53. Blender workspace ejemplo 4.	93
Ilustración 54. Implementación de la cuenta atrás.	95
Ilustración 55. <i>Sensors, Controller & Actuators</i> .	95
Ilustración 56. Blender workspace ejemplo 5.	96
Ilustración 57. Implementación <i>actuator Scene</i> .	97
Ilustración 58. Blender workspace ejemplo 6.	97
Ilustración 59. Implementación <i>actuator parent</i> .	98
Ilustración 60. Implementación <i>actuator game</i> .	99
Ilustración 61. Ventana MENÚ PRINCIPAL.	100
Ilustración 62. Ventana ELEGIR NIVEL.	101
Ilustración 63. Ventana AYUDA I.	102
Ilustración 64. Ventana AYUDA II.	102
Ilustración 65. Ventana CRÉDITOS.	102
Ilustración 66. Ventana GAME OVER.	103

Ilustración 67. Ventana WINNER.	103
Ilustración 68. Archivos necesarios para la ejecución del juego.	104
Ilustración 69. Curva de aprendizaje.	108

Tablas

Tabla 1. Clasificación tipos de motores de juego.	44
Tabla 2. Herramientas de modelado.	52
Tabla 3. Motores de juego.	52
Tabla 4. R1 Tecnología 3D.	54
Tabla 5. R2 Plataforma en la que se ejecutará.	54
Tabla 6. R3 Sonido.	55
Tabla 7. R4 Controles.	55
Tabla 8. R5 Interfaz de usuario.	55
Tabla 9. R6 Ejecutable.	56
Tabla 10. Requisitos software.	57
Tabla 11. Requisitos del entorno de desarrollo.	57
Tabla 12. Requisitos del entorno de producción.	57
Tabla 13. Estimación inicial.	62
Tabla 14. Objetivos usabilidad y jugabilidad.	106
Tabla 15. ERROR 1.	116
Tabla 16. ERROR 2.	116
Tabla 17. ERROR 3.	116
Tabla 18. ERROR 4.	116
Tabla 19. ERROR 5.	117
Tabla 20. SATISFACCIÓN en menores de 30 años.	118
Tabla 21. APRENDIZAJE en menores de 30 años.	119
Tabla 22. EFECTIVIDAD en menores de 30 años.	120
Tabla 23. INMERSIÓN en menores de 30 años.	121
Tabla 24. MOTIVACIÓN en menores de 30 años.	122
Tabla 25. EMOCIÓN en menores de 30 años.	123
Tabla 26. Evaluación general de las características para personas menores de 30 años.	124
Tabla 27. SATISFACCIÓN en mayores de 30 años.	125
Tabla 28. APRENDIZAJE en mayores de 30 años.	126
Tabla 29. EFECTIVIDAD en mayores de 30 años.	127
Tabla 30. INMERSIÓN en mayores de 30 años.	128
Tabla 31. MOTIVACIÓN en mayores de 30 años.	129
Tabla 32. EMOCIÓN en mayores de 30 años.	130
Tabla 33. Evaluación general de las características para personas mayores de 30 años.	131
Tabla 34. Estimación final.	142
Tabla 35. Diagrama de GANTT final.	143

Gráficas

Gráfica 1. Resultados de la propiedad SATISFACCIÓN	119
Gráfica 2. Resultados de la propiedad APRENDIZAJE.	120
Gráfica 3. Resultados de la EFECTIVIDAD.	121
Gráfica 4. Resultado de la INMERSIÓN.	122
Gráfica 5. Resultados de la MOTIVACIÓN.	123
Gráfica 6. Resultados de la EMOCIÓN.	124
Gráfica 7. Resultados de la SATISFACCIÓN.	125
Gráfica 8. Resultados del APRENDIZAJE.	126
Gráfica 9. Resultados de la EFECTIVIDAD.	127
Gráfica 10. Resultados de la INMERSIÓN.	128
Gráfica 11. Resultados de la MOTIVACIÓN.	129
Gráfica 12. Resultados de la EMOCIÓN.	130
Gráfica 13. Aspectos de la jugabilidad más valorados.	132
Gráfica 14. Aspectos de la jugabilidad menos valorados.	133
Gráfica 15. Aspectos de la jugabilidad a mejorar.	134
Gráfica 16. Valoración general del videojuego.	135
Gráfica 17. Valoración general del videojuego.	135
Gráfica 18. Valoración total.	136

1. INTRODUCCIÓN

La industria de los videojuegos es un valor seguro en la sociedad en la que nos encontramos, no sólo en España sino en todo el mundo. Es uno de los campos que sigue creciendo año tras año, y que además está en continua evolución, lo que permite a los futuros ingenieros, no sólo informáticos, sino también de telecomunicaciones, como es mi caso, y más en la especialidad en la que me he centrado, como es Sonido e Imagen, plantearse un posible futuro laboral dentro de esta industria, con el fin de encontrar un trabajo satisfactorio y con un buen reconocimiento personal.

El desarrollo de proyectos, como es el caso de la implementación de un videojuego, es arduo y costoso. Profesionalmente, están implicadas muchísimas personas y no sólo se caracteriza por la implementación en sí, sino que se trata de un proyecto muy creativo y que imprime una gran motivación para las personas que lo realizan.

En este proyecto, veremos principalmente cómo se desarrolla un videojuego con la herramienta *Blender*. Sin embargo, he querido darle un enfoque distinto mostrando las diferentes fases en la realización de éste, desde un punto de vista más profesional. Podremos ver, como un cliente ficticio nos encarga el trabajo y nos transmite los requisitos que deben ser incluidos en el proyecto, y como nos enfrentamos a ellos para poder obtener la solución que más se adecúe a nuestra capacidad y presupuesto. También, observaremos la realización de una fase de pruebas llevada a cabo con personas de diferentes edades, para detectar los fallos del videojuego y así corregirlos. Y por último, veremos los resultados obtenidos a través de una fase donde mediremos la jugabilidad del proyecto, sometiendo a las personas que han probado el videojuego a un test para saber cuáles son los puntos fuertes del mismo, con el fin de potenciarlos, y cuáles los puntos débiles para poder mejorarlos.

A continuación, se explica, brevemente, los apartados en los que está definida la memoria y un resumen sobre lo que se expone en su interior:

- **Objetivos del proyecto.** Se desarrollan los diferentes objetivos que queremos alcanzar con la realización del proyecto.
- **Estado de la cuestión.** Veremos una panorámica sobre los dos aspectos principales del proyecto, el modelado 3D y los motores de juego, para entender la evolución en ambos campos y donde se encuentran en la actualidad.

- **Análisis.** Partiremos de unos requisitos establecidos por un cliente ficticio para así, desglosar el proyecto en tareas y determinar las posibles soluciones, con el fin de obtener la solución óptima.
- **Desarrollo del proyecto.** Explicaremos la fase de modelado 3D del videojuego, su implementación y su post-producción.
- **Fase de testing.** Mostraremos una pequeña parte del estudio al que se someten todos los videojuegos antes de su salida al mercado y estudiaremos los resultados obtenidos a partir de la realización de un test de jugabilidad y de pruebas sobre un conjunto discriminado de personas en función de su edad.
- **Líneas futuras.** Expondremos las posibles líneas futuras que podemos desarrollar a partir de este proyecto.

1.1. OBJETIVOS DEL PROYECTO

El objetivo principal de este proyecto es diseñar y desarrollar la versión beta de un videojuego con la herramienta Blender en su versión 2.49b. Se trata de un videojuego, en el que el jugador se encontrará con tres niveles, que irán incrementando en dificultad según los vaya superando. Para ello contará con diversas habilidades que le ayudarán a la hora de conseguir el objetivo final y retos que tendrá que ir superando, y que le harán más complicado el camino hasta dicho objetivo.

A su vez, se pretende realizar, por un parte un estudio sobre la jugabilidad, aspecto muy importante en la creación de un videojuego que no se ha tenido en cuenta en proyectos similares. Entendiéndolo, no sólo de manera teórica, sino también de manera práctica mediante la realización de un test. Y por otra, construir una panorámica sobre lo que se encuentra detrás de un diseño de esta envergadura, del esfuerzo no sólo del desarrollo en sí, sino del tiempo empleado en el aprendizaje de la herramienta con la que se ha realizado, y en la exigencia creativa necesaria para el diseño de cada nivel y de los retos, intentando ser original y creativo, sin ser redundante.

Teniendo en cuenta los objetivos anteriores, se pretende hacer un proyecto fin de carrera diferente y no necesariamente relacionado con la carrera estudiada, sino disfrutando y aprendiendo una nueva rama de la ingeniería, en este caso la ingeniería informática, que he encontrado fascinante, frustrante y a la vez maravillosa; y saber que en el futuro seré capaz de conseguir cualquier cosa que me proponga, y que no estoy limitada única y exclusivamente a los conocimientos adquiridos durante mi formación.

Y por último, formar parte, aunque de manera muy discreta, de este gran gigante que nos fascina a muchos, que seguimos diariamente, y que parece no tener límite.

2. ESTADO DE LA CUESTIÓN

El desarrollo de un videojuego es un proyecto que precisa de gran creatividad, de un amplio conocimiento sobre las diferentes etapas para su realización y las herramientas que disponemos para ello.

Hoy en día, en un proyecto de esta envergadura se implican muchísimos aspectos artísticos, tecnológicos y económicos. Sin embargo, hay dos en concreto que son los pilares básicos en un desarrollo como este, por lo que se precisa de un estudio muy exhaustivo de ambos. Estos campos son, el modelado 3D y la implementación de la lógica del videojuego mediante un motor de juego.

En el modelado 3D es muy importante la creatividad del desarrollador, mediante esta técnica se diseñan tanto los escenarios del videojuego como los personajes que van a formar parte de estos. Con el modelado 3D, es posible crear mundos fantásticos de gran calidad gráfica, consiguiendo fuerte realismo en las escenas con un impacto visual percibido en los jugadores de manera notable.

Por otro lado, la implementación lógica mediante un motor de juego, es lo que permite que el jugador tenga una interacción con el escenario creado mediante el modelado 3D. En esta fase, se desarrollan los movimientos de los personajes, así como la mecánica del juego.

Tras esta pequeña introducción sobre la importancia que tiene hoy en día estos dos aspectos, vamos a ver cómo han ido evolucionando a lo largo de la historia, y en qué momento, su unión ha hecho que el mundo de los videojuegos sea uno de los desarrollos más creativos y rentables que hay en la actualidad. Una de las cuestiones más importantes que vamos a ver a continuación, es que indirectamente la evolución de la tecnología computacional ha ido marcando la evolución de ambos aspectos.

2.1. MODELADO 3D.

El modelado 3D es un proceso que se realiza por ordenador, mediante el cual se diseñan figuras, objetos y escenas, con el fin de elaborar proyectos como películas de animación, efectos especiales para películas o videojuegos. El hecho de que una animación se diseñe en 3D significa que los objetos están modelados en un espacio tridimensional, donde cada uno de ellos viene definido por las coordenadas X, Y, Z, o lo que es lo mismo, por su anchura, longitud y altura, lo que determinan su existencia en dicho espacio.

El modelado 3D hace que la percepción de los objetos diseñados mediante esta técnica sea muy similar a la que una persona experimenta día a día, con lo que, las

escenas que podemos crear a partir de dichos objetos trasladan al sujeto a un mundo realista, incrementando así las sensaciones y emociones percibidas, frente a las escenas que se habían diseñado, hasta el momento, realizadas en 2D.

En el siguiente punto, vamos a ver como la animación 3D se introduce poco a poco y de manera aislada, sobre todo en películas de ficción, para más tarde realizar proyectos de animación donde todas sus escenas están realizadas utilizando esta técnica.

2.1.1. Cronológicamente.

En este apartado vamos a recorrer de manera breve la historia del diseño en 3D. Veremos que el inicio del modelado 3D se produjo en la industria del cine, y que gracias a las inquietudes de los productores y directores del séptimo arte, se ha podido aplicar la tecnología del modelado para más campos.

- **Década de los 70 y los 80.** Durante estos años, el diseño de imágenes por ordenador comenzaba a introducirse en películas de ficción y de animación. Se realizaban diseños de muy baja calidad pero que permitían la creación de efectos visuales no conocidos hasta ese momento.
 - En 1973, *Michael Crichton* escribió y dirigió “*WestWorld*”, esta fue la primera película que utilizó gráficos diseñados por ordenador y una de las primeras en tratar el tema de la rebelión de las máquinas. La empresa encargada del diseño fue “*Triple-I*”[1] y las imágenes generadas eran en dos dimensiones.

Más tarde, en 1976, el director *Richard T. Heffron*, hizo la secuela de “*WestWorld*” llamada “*Future World*”. Esta película fue la primera en la que se introdujeron imágenes en 3D diseñadas por ordenador. Se utilizaron para animar las manos y la cara de los robots.

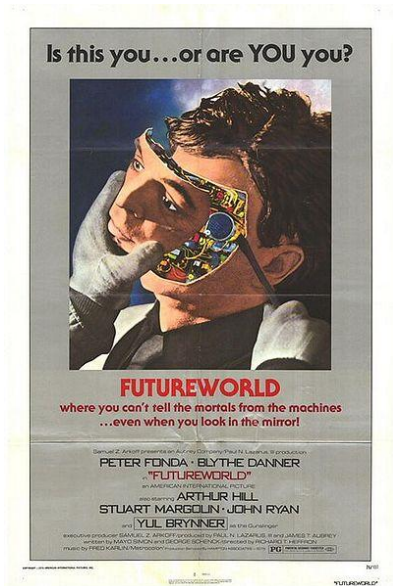


Ilustración 1. Cartel de la película FutureWorld.

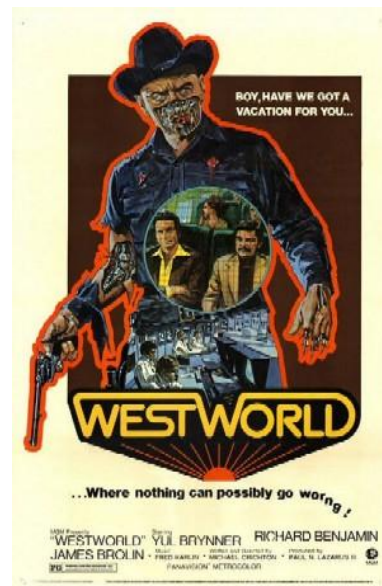


Ilustración 2. Cartel de la película WestWordl.

- En 1981, se lanza la película de animación *"Adam Power and the Jungle"*, creada por Adam Power, se trataba de una animación en la que un malabarista realizaba juego malabares con diversos objetos. Fue una de las primeras animaciones realizadas en 3D y la empresa encargada de ello fue, como en la película *"WestWordl"*, la empresa tecnológica *"Triple-I" (Information International Inc)*.



Ilustración 3. Imagen de la película " Adam Power and the Jungle".

- En 1982, *Walt Disney* lanza la mítica película *"Tron"* que revoluciona el mundo de la computación gráfica. En esta película un programador se introduce dentro de los circuitos de un ordenador donde se encuentra que los programas tienen vida y donde se enfrenta a diversos peligros.

Para el diseño de los gráficos, la empresa *Disney* compró la máquina PDP-10 fabricada por DEC (*Digital Equipment Corporation*)[2], uno de los ordenadores más potentes del momento.

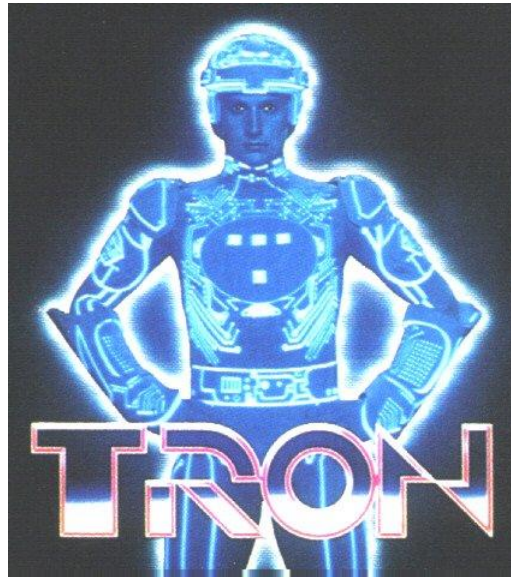


Ilustración 4. Imagen de la película "Tron".

- **Años 90.** En esta década se empiezan a integrar los diseños 3D, no sólo en películas de animación sino en todo tipo de películas, entendiendo una nueva forma de realizar los efectos especiales. Además, las películas de animación comienzan a desarrollarse íntegramente por ordenador. Esta década supone una revolución, ya que a partir de entonces, los efectos especiales se harán de forma computacional.
 - En 1991, la empresa *Industrial Light & Magic*[3] ayudó al director *James Cameron*, en su película *Abyss*, a crear mediante la computación gráfica una de las escenas más recordadas de esta película, fue una de las primeras veces que se introducía la animación diseñada por ordenador de una manera realista.

Otra de las escenas que supuso una revolución en este campo es la que podemos apreciar en la ilustración 5 de la película *Terminator 2*.

Más tarde, en 1997 James Cameron volvería a sorprendernos con unos maravillosos efectos especiales, por supuesto realizados por ordenador por la compañía *Digital Domain*[4], en su espectacular y premiada película "Titanic".



Ilustración 5. Imagen "Terminator 2".



Ilustración 6. Imagen de la película "Titanic".

- En 1993, la película "*Jurassic Park*" del director *Steven Spielberg*, fue una de las primeras en mostrar un grandísimo resultado mezclando escenas en "carne y hueso" con imágenes generadas por ordenador. El resultado de la combinación entre el movimiento de los dinosaurios mediante bombas neumáticas y los retoques del resto de los movimientos mediante el modelado 3D, como en el caso de la persecución protagonizada por el T-Rex, queda patente hoy en día. La empresa encargada de semejante proeza fue *Industrial Light & Magic*, la misma empresa encargada de los efectos de la *Abyss*.



Ilustración 7. Imagen película "Jurassic Park".

- En ese mismo año, la animación 3D llegó al mundo de la publicidad. La empresa, mundialmente conocida, *Coca-Cola*, lanzó un revolucionario spot para la campaña de navidad, íntegramente diseñado por ordenador por el estudio de animación en 3D *Rhythm & Hues*[5].



Ilustración 8. Imagen del spot publicitario de Coca-Cola.

- En 1995, la empresa Disney y la que era en ese momento una de sus factorías Pixar[6], crean la primera película de animación diseñada en su totalidad mediante modelado 3D, la famosa "*Toy Story*" y que supone el comienzo de grandes producciones gráficas. Entre estos proyectos, se encuentra su segunda parte "*Toy Story 2*" que vio la luz en 1999, y que ofrecía una mayor calidad en las imágenes y una mayor limpieza y realismo en los movimientos de los personajes.



Ilustración 9. Cartel de la película "Toy Story".



Ilustración 10. Imagen de la película "Toy Story" .

- En 1996, el mundo de los videojuegos y de la animación 3D se encuentran por primera vez en el videojuego Quake.



Ilustración 11. Imagen videojuego Quake.

- **Años 2000.** Durante estos años, los efectos especiales ya no se conciben de otra manera que no sea mediante el diseño por ordenador. Gracias al aumento de la tecnología de estos equipos, la calidad de los efectos es espectacular y claramente patente en los proyectos citados a continuación.
 - En el año 2000, hay que destacar la escena, realizada íntegramente por ordenador, de la película “El hombre invisible”, en ella el espectador admiraba como el protagonista iba sufriendo dicha transformación poco a poco, viendo claramente como la piel, los músculos y los huesos iban desapareciendo.

- Más tarde en el 2001, *Peter Jackson* nos sorprendió a todos con la adaptación al cine de la, mundialmente conocida, trilogía de *J.R.R. Tolkien* “El señor de los anillos”, con “La Comunidad del Anillo” en 2001, “Las dos Torres” en 2002 y “El Retorno del Rey” en 2003, con unos efectos especiales realizados por ordenador por las compañías *WETA* y *Digital Domain*, y sin los cuales el mundo creado por *Tolkien* no se hubiera entendido de la misma manera.



Ilustración 12. Escena de la película "La Comunidad del Anillo"

Vemos que en estos años, prácticamente todas las películas introducen en sus escenas animación en 3D, y es realmente difícil encontrar alguna superproducción que no la tenga.

Hay que destacar, que estamos viviendo hoy en día una nueva revolución en la animación 3D, hasta el punto de rodar una producción íntegramente en 3D, y como no, uno de los genios del cine y del que hemos hablado anteriormente, James Cameron, es el pionero de esta nueva concepción de entender el cine, rodando en el año 2009 la película Avatar.



Ilustración 13. Escena de la película "Avatar".

A continuación, vamos a ver gráficamente la evolución del modelado 3D desde los años 70 hasta la actualidad.

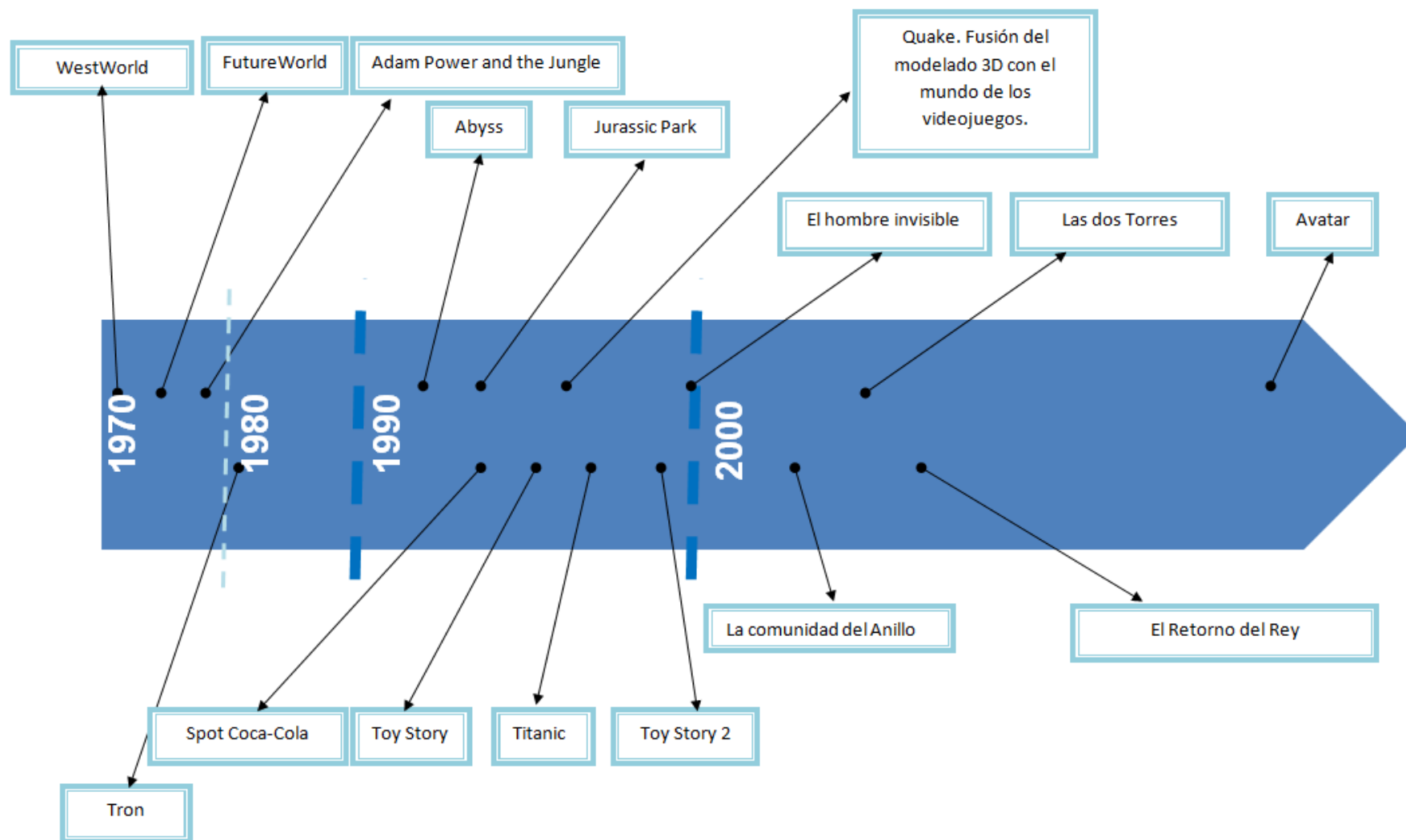


Ilustración 13. Eje cronológico del los proyectos que incluyen animación por ordenador hasta la actualidad.

2.1.2. Herramientas de modelado 3D.

Hoy en día, tenemos a nuestra disposición multitud de herramientas de modelado. Conociendo las necesidades que debemos cubrir para la realización de nuestro proyecto, debemos analizar cada una de las herramientas, para así elegir la que más se adapte a los objetivos que queremos cumplir.

A continuación vamos a detallar las características de algunas de las herramientas disponibles en el mercado, así como los proyectos realizados con ellas.

- **Autodesk MAYA[7].**

Aplicación de modelado 3D, que dispone de herramientas de tratamiento de objetos mediante mallas, subdivisión de polígonos y modelado de superficies. También nos ofrece herramientas de texturizado, pintado de vértices y *unwrapping* con imágenes. Además, podemos animar nuestros modelos 3D, con posibilidad de realizar un tratamiento físico de partículas, fluidos y nubes, entre otras cosas. También dispone de renderizado y tratamiento de luces y sombras, así como de una herramienta de colocación de cámaras.

Su interfaz gráfica puede ser customizada al gusto del usuario, dispone de atajos de teclado y de ratón, así como multitud de menús y de botones con sus opciones correspondientes. Debido a su complejidad, es poco intuitiva y requiere de un tiempo de aprendizaje alto.

Se trata de una aplicación de pago (3499\$), multiplataforma, podemos utilizarla en los sistemas operativos *Windows*, *OSX* y *Linux*, y con una compatibilidad con los *API OpenGL*, *DirectX* y *Software Rendering*. Con esta herramienta se han realizado multitud de proyectos, tanto cinematográficos, como de la industria del videojuego, entre ellos el videojuego "*Medal of Honor*".

- **SOTFIMAGE[8].**

Esta herramienta nos proporciona un entorno de creación interactivo de objetos 3D, con la posibilidad de tratar los objetos de forma única o divididos en superficies o incluso en polígonos divididos. Nos proporciona un editor de flujos de trabajo para planificar de forma ordenada nuestro proyecto, guardándolo en un archivo independiente todos los días que

trabajemos con el programa. Además nos proporciona la animación de nuestros modelos, incluyendo un tratamiento facial especial, y el tratamiento de la física, incluyendo partículas y pelo, entre otras cosas.

También dispone de un editor de texto, para poder realizar nuestra propia implementación con su propia SDK y API.

La interfaz gráfica que nos ofrece dispone de barras de herramientas, con atajos de teclado y de ratón para cada una de las opciones que podemos utilizar, al igual que multitud de menús. A pesar de todo, es poco intuitiva por lo que su curva de aprendizaje es alta, sin embargo, podemos obtener unos buenos resultados una vez que el usuario se familiariza con la interfaz.

Se trata de una aplicación de pago (2.999\$), por eso es muy utilizada por grandes factorías que han realizado proyectos muy ambiciosos en varios campos. A destacar, el videojuego “*Metal Gear Solid 4*”.

- **3D STUDIO Max[9].**

Se trata de una herramienta de modelado 3D muy utilizada por diseñadores gráficos y desarrolladores de videojuegos para PC, debido principalmente a que únicamente se puede desarrollar en una máquina que disponga del sistema operativo. A parte de ofrecernos prácticamente todas las herramientas incluidas en los mejores editores de modelado 3D, contamos con lo que sus creadores denominan “*Nitrous Accelerated Graphics Core*”, esta característica adicional aumenta la velocidad de iteración, con lo que realiza las operaciones de modelado bastante más rápido que otras aplicaciones del mercado, y además, maneja un gran número de datos más interactivamente, permitiendo realizar el modelado de polígonos a gran velocidad.

También hay que destacar, la técnica de renderizado artístico, pudiendo representar gráficamente los escenarios creados en diversas técnicas pictóricas. Por supuesto, incorpora animación y simulación de partículas y fluidos, entre muchas otras, y tratamiento de luces y sombras.

Otra característica importante a destacar de esta herramienta es la integración y externalización en multitud de formatos, lo que le da al desarrollador una mayor flexibilidad a la hora de trabajar. Incluye su propia librería de objetos y funciones 3ds Max SDK. Además, dispone de una

interfaz muy completa e intuitiva a la vez, con lo que el tiempo de aprendizaje es menor que con otras herramientas.

Al igual que las anteriores, se trata de un software de pago (3495\$). Algunos de los proyectos realizados con esta herramienta son los videojuegos “*Assassin’s Creed*” o “*Halo*” de gran repercusión mediática.

- **LIGHTWAVE**[10].

Esta herramienta nos ofrece la mayoría de las técnicas de modelado que las anteriores vistas, pero con algunos inconvenientes, aparte de que también se trata de una herramienta de pago (795\$), aunque significativamente menor, es necesario añadir *plugins* que se adquieren de manera independiente para poder obtener los mismo resultados.

Un punto a su favor es la integración y externalización en multitud de formatos, y además, que dispone de una interfaz gráfica intuitiva y bastante sencilla de aprender.

En cuanto a los requisitos hardware, el desarrollador debe disponer o bien de una máquina con sistema operativo Windows o con OSX. Es compatible con el API OpenGL, y entre los proyecto desarrollados con él se encuentra el videojuego “Torque”, que como veremos más adelante, se realizó con el motor de juego del mismo nombre. Con esto, esta aplicación demuestra que se puede realizar un gran proyecto.

- **BLENDER**[11].

Se trata de una aplicación de modelado 3D, que también incorpora su propio motor de juego, que nos ofrece multitud de herramientas de modelado, con una variedad de objetos muy amplia y con posibilidad de hacer grandes desarrollos escultóricos.

Dispone de un tratamiento de animación casi profesional, con posibilidad de simulación de partículas, pelo, ropa y fluidos, entre otras muchas. Texturizado, pintado y unwrapping, además de edición de audio y video, e incorpora un editor de texto para realizar scripts en Python a la hora de implementar en su motor de juegos. También integra y externaliza sus proyectos en multitud de formatos, entre ellos y a destacar, en el formato de la herramienta 3d Studio. Se trata de una herramienta multiplataforma, ejecutable en cualquier sistema operativo, con compatibilidad con OpenGL.

Una de las características principales en cuanto a la implementación en Blender es que se trata de una implementación visual. Y la más importante de todas, es que se trata de un proyecto Open Source, con lo que su adquisición es gratuita, así como su documentación. El mantenimiento de la herramienta la realiza Blender Foundation. Entre los proyectos realizados se encuentran multitud de cortos y un videojuego para PC “Yo Frankie!”.

- **CINEMA 4D[12].**

Es una de las herramientas más avanzadas en modelado 3D, debido a sus técnicas en la creación de personajes, gracias a su potente simulador de partículas y pelo, así como de su texturizado y pintado.

La hace más interesante el hecho de que su interfaz gráfica tenga una curva de aprendizaje media-baja. Sin embargo, debido a su calidad en el modelado, se trata de una de aplicaciones más caras que hay en el mercado (3695\$).

Las plataformas en las que se puede ejecutar son Windows y Linux, y dispone de compatibilidad OpenGL. Es utilizado en animación para publicidad y también para simulación médica.

- **HOUDINI[13].**

Se trata de una aplicación muy potente en la animación y modelado de objetos 3D. Dispone de una gran variedad de tipos geométricos, de una herramienta para el tratamiento de músculos y de la piel humana, y de técnicas para el manejo de luces y sombras dentro de la escena. Además, nos ofrece un editor de animación con tratamiento por frames y por canales de video diferenciados. Dispone de una herramienta para realizar diagramas de flujo de nuestro proyecto, editable en función de su desarrollo.

Es una de las aplicaciones más económicas del mercado (1.995\$), multiplataforma y con compatibilidad con OpenGL. Uno de los proyectos más conocidos realizados con esta herramienta es el videojuego “KillZone 3”.

- **RHINOCEROS**[14].

En la línea de HOUDINI, esta herramienta proporciona modelado de objetos con el tratamiento de curvas NURBS, de superficies y de sólidos de manera casi ilimitada. La calidad gráfica del modelado 3D es prácticamente igual que los productos que le superan en precio (995€). Nos ofrece el modelado de objetos tanto en 3D como en 2D, pudiendo bascular entre ellas de manera sencilla, de ahí que sea utilizado sobre todo en el campo de la ingeniería. Gran compatibilidad de archivos con otros formatos, además de poder introducir diversos clientes de renderizado.

Su interfaz gráfica es sencilla y con una curva de aprendizaje baja. Es muy utilizado en el prototipado y diseño de vehículos.

- **CHEETAH 3D**[15].

Se trata de una herramienta sólo disponible para el sistema operativo OSX, ofrece una herramienta de modelado 3D con multitud de polígonos, además de operaciones booleanas, subdivisión de superficies y el tratamiento de mallas. También dispone de texturizado y pintado de objetos y unwrapping con imágenes. En cuanto a la animación, sólo dispone de animaciones simples y una herramienta específica para animación facial. Una de las cosas de las que carece es de renderizado .

Su interfaz gráfica es potente, intuitiva y rápida de aprender. Por eso, es uno de los más utilizados para proyectos de bajo presupuesto, o para animaciones amateur, además de por ser bastante económico (149€) pero limitado en cuanto a los requerimientos hardware

- **ZBRUSH**[16].

Herramienta de modelado 3D muy potente, capaz de realizar tratamientos de objetos tanto en 2D como en 3D, de una calidad impresionante. Dispone de multitud de herramientas para esculpir objetos de manera muy precisa, ya que únicamente nos ofrece el tratamiento de superficies y polígonos, pero a un nivel muy superior al de las otras herramientas. El texturizado y pintado se puede hacer en numerosas capas, con lo que esta función se puede realizar de manera minuciosa. Sin embargo, no dispone de unwrapping de imágenes. Carece de herramienta de animación, pero se pueden integrar sus modelos a alguna herramienta externa para proceder a su animación.

Tiene un interfaz bastante tedioso debido a la complejidad de las herramientas y al hecho de no contar con un editor de animación, hace que el usuario deba aprender a utilizar como mínimo dos herramientas distintas.

Se trata de una herramienta multiplataforma y de pago (489\$), se utiliza para el desarrollo de anuncios y efectos especiales en el cine.

2.2. MOTORES DE JUEGO.

El término motor de juego (“Game Engine”) nació a mediados de la década de los 90. Su principal objetivo es abstraer las tareas comunes en el desarrollo de videojuegos como son, la representación o la física, para que los desarrolladores puedan centrarse en detalles específicos, que son los que diferenciarán un juego de otro.

Un motor de juego es una aplicación que permite implementar un videojuego, proporcionando rutinas de programación para su diseño, creación y representación. Su funcionalidad básica es aportar un renderizado de gráficos en 2D y 3D, en el que se incluye un motor físico o detector de colisiones, sonidos, scripting, animación, inteligencia artificial, redes, *streaming*, administración de memoria y un escenario gráfico.

La mayoría de los motores proporcionan un API dentro de su SDK que permite establecer una comunicación directa con las herramientas descritas en el párrafo anterior.

El nacimiento de los motores contribuyó al desarrollo de los primeros juegos en 3D llamados “first-person shooters” FPS, es decir, juegos donde el jugador desempeñaba el papel de un tirador, y donde la vista es en primera persona, dándole un mayor realismo a la historia del videojuego. De manera que se considera que el primer videojuego desarrollado mediante un motor de juego es Doom, y supone el desencadenamiento del crecimiento de una de las industrias que más facturación obtienen en todo el mundo.

El hecho de que el argumento de los videojuegos haya variado a lo largo de los años, está relacionado a que el perfil, para el que están destinados, va evolucionando por lo que demandan un enfoque distinto al de los videojuegos tradicionales (FPS) que había hasta el momento, sumado a que el entorno de esta industria crece mano a mano con la tecnología asociada a ella, hace que el desarrollo del proyecto de un

videojuego vaya aumentando en complejidad y que surjan empresas que se especialicen en un público y un estilo determinados.

La evolución en los motores hace que estos sean más complejos en contraposición con lo que el usuario final aprecia. El refinamiento continuo en los motores ha creado una fuerte separación entre *rendering* (representación), implementación, trabajo creativo y el nivel de diseño. El desarrollo mediante hilos tomó más importancia a raíz de los sistemas *multi-core* y el incremento de la demanda de realismo. La clave de la evolución es el desarrollo en múltiples plataformas.¹

2.2.1. Cronológicamente.

Se debe destacar principalmente dos factores, encargados de definir la evolución de los motores de juego. Uno es el avance tecnológico de los PC, puesto que, a medida que han ido apareciendo nuevas tecnologías hardware incrementando la potencia de la máquinas, se han podido crear motores más especializados y sofisticados capaces de aprovechar al máximo las nuevas cualidades de las CPU's. Al ser un proyecto con el que se desarrollan productos, la rentabilidad de estos es muy importante y a la vez las restricciones en su creación. Restricciones tales que el desarrollo tecnológico ha ido disminuyendo con los años.

El segundo factor es la manera de enfocar el proyecto. Como se verá a continuación, durante los primeros años, los motores estaban ligados a los juegos que se desarrollaban con ellos, estableciendo la tendencia a realizar con un motor un único videojuego. Según fue avanzando el tiempo, los motores se empezaron a considerar como un producto integrado dentro del desarrollo empresarial. El motivo fue que el coste de realizar un motor nuevo cada vez que se desarrollaba un videojuego, aumentaba considerablemente y los desarrolladores no podían hacer frente a semejante coste. Cada vez más empresas se especializaban en el desarrollo de motores más completos, dejando a un lado la creación de juegos, con el fin de comercializar el producto a las nuevas empresas que habían surgido encargadas de la implementación de videojuegos.

Por tanto, se produjo una separación de la industria; empresas especializadas en el desarrollo de los motores, llamadas *middelware*, y empresas especializadas en el desarrollo de los videojuegos, siendo éstas las que deciden si su producto se comercializará o si estará dentro de un proyecto en código abierto disponible para cualquier empresa o persona.²

A pesar de que el desarrollo de los motores de juego no empezó como tal hasta principios de los 90, hay que tener en cuenta que los descubrimientos tecnológicos

¹ Bibliografía: 1.

² Bibliografía 1

y los videojuegos que nacieron durante los años 70 y 80 forman parte del nacimiento de los motores.

Por todo ello, a continuación, se explica brevemente los avances que supusieron un cambio en el enfoque del desarrollo de los videojuegos y que hicieron que llegásemos a la tecnología que hoy en día podemos usar para su creación.

- **Años 70 y 80.** Durante estos años nacieron los juegos que se basaban en el concepto de perspectiva en primera persona, llamada comúnmente FPS. Estos videojuegos situaban al usuario en una nueva forma de jugar, en una perspectiva totalmente diferente a la de juegos anteriores donde el personaje se movía en un mundo de plataformas 2D, con una calidad gráfica bastante pobre. Esta nueva forma de jugar, infligía más realismo y hacía que el jugador se sintiera más identificado con su personaje virtual. Los gráficos no simulaban una visión de la realidad tal y como ahora la conocemos, pero sí reflejaban de manera más fiable los escenarios representados. Para el desarrollo de estos videojuegos y debido a la necesidad de disparar a objetivos concretos, hizo que los desarrolladores usaran motores de vectores gráficos.
- **Principios de los 90.** La revolución de los videojuegos y de su desarrollo, llegó con Doom 2.5D, aunque todavía no debería incluirse en la generación de juegos en 3D, puesto que los gráficos eran planos y los objetos sprites, sin embargo, nunca antes se habían visto unos gráficos con esa calidad. En cuanto a la tecnología necesaria para la ejecución de los videojuegos que existían en el mercado durante estos años, se necesitaban unos requerimientos medios en hardware de video, ya que el renderizado era una operación que tenía que realizar íntegramente la CPU, lo que suponía una carga computacional bastante grande y por tanto una máquina que no todas las personas podían permitirse.
- **Mediados de los 90.** En estos años sí podemos decir que nacieron los primeros videojuegos en 3D. Se recreaban mundos mediante niveles de geometría arbitrarios. En lugar de la utilización de sprites, se empezaron a usar objetos hechos a través de polígonos texturizados. Los requerimientos hardware aumentaron debido a la calidad gráfica del renderizado, y por primera vez se utilizaron aceleradores 3D. Sin embargo, a pesar de que la evolución de las máquinas era rápido, la capacidad de las tarjetas de video seguía siendo insuficiente, por lo que parte del renderizado seguía realizándolo la CPU. A todo ello hay que sumarle que no existían plataformas gráficas realmente estables y que, por tanto, el desarrollo de los videojuegos no era independiente de éstas.

- **Finales de los 90.** En este periodo *Nvidia*[17] fabricó su primera tarjeta gráfica que incluía el *hardware T&L*, la famosa *GeForce 256*, lo que supuso la evolución de VGA a GPU, abriendo una nueva perspectiva en el desarrollo gráfico de videojuegos. Muchas de las empresas del momento no pudieron competir. Microsoft, que despuntaba con DirectX 7, pudo mantenerse a flote al permitir acceder al T&L de manera sencilla. ATI[18], que inició su incursión en el mundo de las tarjetas gráficas, se convirtió en su competidor más serio. Esta nueva tarjeta gráfica se encargaba de manejar la iluminación y parte de la geometría por sí sola, disminuyendo así la carga computacional que tenía que soportar la CPU en los años anteriores. Uno de los primeros motores que se beneficiaron de este avance fue el Unreal Engine, uno de los más conocidos y que más rentabilidad ha obtenido con la comercialización de sus juegos de perspectiva en primera persona.
- **Principios del 2000.** Las nuevas tecnologías hardware de los años anteriores, posibilitaron que gráficamente aumentasen los detalles y que los entornos exteriores fueran de mayor calidad, haciendo que el concepto “*rag-doll physics*” tuviera un gran papel. Hoy en día no concebimos un juego en el que el viento, el fuego o los fluidos no estén bien representados, pero hasta esta época no era posible, momento en el que la física de partículas comenzó a despuntar. Disfrutar de estos videojuegos requería el uso de las nuevas tarjetas gráficas, y disponer de una CPU poderosa no nos libraba de dichas limitaciones.
- **Mediados del 2000.** La lucha entre ATI y Nvidia hace que las tarjetas vayan evolucionando a pasos agigantados. Además, Microsoft[19] saca al mercado su nueva joya, la XBOX 360. Esto desemboca en una mayor calidad gráfica, con lo que los entornos toman mucha importancia, tanto interiores como exteriores, y el desarrollo con los nuevos motores lo hace posible, puesto que incluían un tratamiento muy eficaz de la iluminación, física *shader*, *bump mapping* y la técnica *shader* en vértices, creando así gráficos de muy alta calidad.

Por tanto, esto hace que, gráficamente, los videojuegos sean mucho más potentes y de esta manera, más atractivos para el usuario. A partir de este momento, las videoconsolas de nueva generación marcan el curso de los motores de juego.

- **Finales del 2000.** En estos años, los desarrolladores persiguen una calidad fotográfica mucho más realista. La consiguen explotando fuertemente los recursos descubiertos en años anteriores y añadiendo algunos nuevos, provenientes de otras industrias, tales como la cinematográfica (profundidad de campo, desenfoque de movimiento, etc.). Y es que a partir de 2007, los

motores de juegos no presentan grandes cambios, tan solo van mejorando pequeños aspectos en función de la necesidad. Con la variedad disponible, se pueden hacer verdaderas obras de arte, en lo que a la industria del videojuego se refiere. El hecho de que se haya producido esta desaceleración, se debe a que los gigantes de las videoconsolas son los que marcan las pautas de crecimiento en el sector.

Por tanto, el avance tecnológico de los motores de juego ha sido sobresaliente en los quince primeros años a partir de “Doom”, ligado, indiscutiblemente a los avances en el software y en el hardware de las máquinas, que ha ido creciendo exponencialmente durante este tiempo. Todo ello, sumado a la cantidad de dinero que mueve esta industria, hace que el sector esté en un continuo avance hasta alcanzar cotas insospechadas. No hay más que apreciar las nuevas opciones de desarrollo que se han abierto a raíz de las nuevas videoconsolas, por supuesto sin menospreciar el sector del videojuego en PC.³

Una vez visto el desarrollo de las técnicas utilizadas por los motores en general, se debe destacar aquellos que supusieron una revolución para la industria. Como vamos a ver a continuación, muchos de los motores nacieron a raíz del videojuego que sus desarrolladores crearon. Lo que se pretende dejar patente con esto, es que en este mercado un motor de juego puede proporcionarnos multitud de técnicas de desarrollo y puede destacar sobre otros en multitud de aspectos. Sin embargo, sólo serán reconocidos por la repercusión que el videojuego haya tenido.

- **Ultima UnderWorld.** Fue publicado por *Origin System* en 1990. Se trata de uno de los primeros juegos en FPS y del heredero de “*Space Rouge*”. Este motor de juego realizaba un mapeo de texturas mediante un algoritmo que permitía crear planos a distintas alturas e inclinaciones. Los personajes estaban desarrollados mediante figuras en 2D denominadas *sprites*.



Ilustración 14. Ultima Underworld.

³ Bibliografía 2

- **ZZT.** Creado por Tim Sweeney, fundador de *Epic Games*[20], en 1991, era un juego y un motor de juegos para DOS y que en el momento de su lanzamiento ya se encontraba desfasado comparado con los que se comercializaban en ese momento. Sin embargo, debemos mencionar a su creador ya que más tarde revolucionaría el mercado y la industria de los videojuegos con el lanzamiento de *Unreal*.

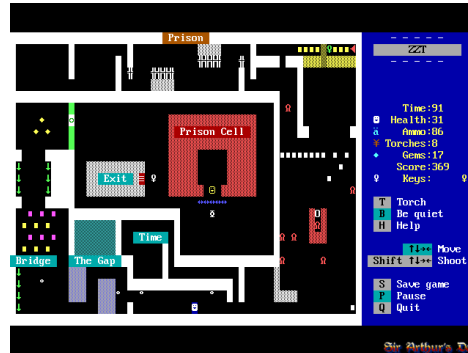


Ilustración 15. ZZT.

- **Wolf3D Engine.** En 1992, John Carmack funda *ID Software*[21] y desarrolla su ópera prima *Wolfenstein 3D*, considerado el primer juego en “3D”. Este motor incluía ray casting para el renderizado de las imágenes y los escenarios, una resolución de 320x200 con 256 colores y paneles de texto superpuesto a los gráficos. Se trataba de un videojuego FPS pero que aportaba una nueva perspectiva haciéndolo más atractivo para el jugador.



Ilustración 16. Wolf 3D.

- **Doom Engine.** Desarrollado también por John Carmack en 1993. A pesar de que la calidad del motor que el mismo lanzó apenas un año antes era pionera en esos años, consiguió aumentar sus prestaciones e implementar el primer videojuego considerado 3D. Supuso una gran revolución de la industria. La calidad de los gráficos no era muy diferente al que se conseguía con el motor anterior y también se ejecutaba sobre DOS, pero su novedad más importante fue el soporte multijugador. Más tarde, *Doom Engine* evolucionó en *Build*

Engine, aportándonos más prestaciones, entre las que se incluían, una mayor resolución de los gráficos y cambios en el tamaño de la pantalla para máquinas menos potentes.



Ilustración 17. Doom.

- **Quake.** Surgió de las manos del mismo creador que desarrolló *Doom*, el lanzamiento de este juego y del motor revolucionó la manera de percibir los videojuegos FPS. Nació en 1996 y entre sus novedades destacaron los *lightmaps*, mapas de luces que se añaden a los objetos estáticos proporcionando una mayor calidad en el renderizado y en lo gráficos, y que hoy en día se continúan utilizando.



Ilustración 18. Quake I.

- **XnGine.** Fue uno de los primeros motores con los que verdaderamente se desarrollaban proyecto íntegramente en 3D, hasta la fecha la capacidad de ofrecer gráficos tridimensionales se basaba en engañar al ojo humano mediante técnicas de luces y sombras aplicadas sobre los objetos. *XnGine* fue

creado por *Bethesda* en 1997 para el desarrollo de uno de sus mejores videojuegos. Introdujo muchos cambios en la industria llegando a variar la forma gráfica de desarrollarlos.

- **IdTech2.** Hasta este momento el desarrollo de un videojuego implicaba el nacimiento de un motor de juego asociado a él. Sin embargo, la creación del motor *IdTech2* por el gran genio John Carmack en 1997, hizo que este concepto cambiara. Con este motor de juego se realizaron grandes éxitos como *Quake2*, *Half Life* o *Counter-Strike*, entre otros.

Las novedades que este motor introducía eran la iluminación coloreada y la aceleración por hardware mediante *OpenGL*[22]. Esto permitió que fuera muy estable y potente a la vez. Además, sus desarrolladores liberaron su código, facilitando que cualquier compañía desarrollara su propio videojuego sin hacer un desembolso en la compra de licencias de otros motores, asegurándose así un alto porcentaje del mercado. A partir de este momento muchas más factorías potenciaron el desarrollo de grandes motores de juego, de esta manera podían realizar grandes proyectos y, a la vez, mantenerlos de sus motores en función de los avances tecnológicos.



Ilustración 19. Counter Strike.

- **Lithtech.** Nació en 1998 mediante la colaboración entre *Monolith*[23] y *Microsoft*. Se caracterizó por dividir la fase de renderizado en dos partes, una mediante tecnología software y la otra mediante *DirectX*[24]. De esta asociación nació *DirectEngine*, lo que más tarde se conocería *Lithtech*, cuando *Monolith* se separó de *Microsoft*. Era un motor rápido y barato, que proporcionaba buena calidad. Estuvo en uso bastante tiempo, hasta que fue rediseñado con el nombre de *EX Jupiter*.
- **Unreal Engine 1.** A raíz del lanzamiento de *IdTech2*, *Tim Sweeney* presenta en 1998 un nuevo motor de juegos, introduciendo el concepto de un gran motor para el desarrollo de diversos proyectos, *Unreal Engine*. El objetivo de este

motor no era revolucionar los gráficos que se podían desarrollar, sino que consiguió que la implementación de éstos fuera mucho más estable. Por tanto, se hizo un “rediseño” de la implementación de todos los aspectos, desde la animación facial, pasando por la percepción de los objetos (iluminación, texturización), hasta la inteligencia artificial, sin menospreciar la “postproducción” (sincronización labial, audio y video). Esta factoría desarrolló uno de los mejores videojuegos para PC de la historia, y es que a nadie que conozca esta industria se olvida de *Unreal*.



Ilustración 20. Unreal.

- **Renderware.** Nació también en 1998 y fue desarrollado por la compañía *Criterion Software*[25]. Lo revolucionario de este motor fue la capacidad de adaptación a la mayoría de plataforma que existían aparte del desarrollo para PC. Hoy en día se sigue utilizando en productos para las nuevas videoconsolas PS2, PS3, Xbox 360. Uno de sus videojuegos más conocidos es "*Grand Theft Auto*".



Ilustración 21. Grand Theft Auto.

- **IdTech 3.** Se lanzó en 1999 y se trataba de una mejora del motor anterior. Entre las actualizaciones, se trató de realizar un motor que fuera estable en red y a su vez optimizado.
- **Unreal Engine 2.** En el año 2000, sale el nuevo *Unreal Engine*. Entre las novedades destacan el desarrollo multiplataforma para las nuevas consolas existentes en el mercado *PS2*, *GameCube* y *Xbox*.
- **Aurora Engine.** Todos los grandes motores descritos anteriormente se encargaban del desarrollo de juegos FPS. Hasta el año 2001 en que nació *Aurora*, ningún gran motor se encargaba de desarrollar otro tipo de videojuegos. Es en este momento cuando renacen los juegos de rol en formato digital. Lo importante de este motor no es su potencia gráfica sino la forma que tiene de generar los entornos. Éstos se generaban mediante un editor de niveles muy sencillos que sin embargo desarrollaba espacios visualmente más que aceptables.
- **Torque.** Este motor desarrolla proyectos íntegramente en 3D. Fue creado para el lanzamiento del videojuego *Tribes 2*, uno de los más relevantes en FPS. Una de sus características es que modelaba los entornos manipulando los niveles de manera dinámica, disminuyendo la cantidad de polígonos a renderizar. También destacó al incluir un editor de texto y de interfaces gráficas mediante la técnica *drag-and-drop*. Además, disponía de código flexible en red para la opción multijugador. Por todo ello, se convirtió en uno de los motores más robustos a pesar de no tener grandes títulos detrás de él.



Ilustración 22. Tribes 2.

- **Odyssey Engine.** Se trata de la continuación del motor *Aurora* y nace en 2003, su novedad es el desarrollo de espacios tridimensionales. La tecnología ya

avanzada en estos años hace que se desarrollen grandes juegos con este motor como fue *Los Caballeros de la Antigua República I y II*.

- **Id Tech 4.** *Id Software* continúa mejorando su motor de juegos. Destaca la utilización de los mapas mediante *bump mapping*, *normal mapping* y *specular highlight*. Estas técnicas combinadas hacen que el desarrollo de los espacios sea de una calidad gráfica realmente espectacular. Sin embargo, los requerimientos hardware eran bastante altos lo que supuso que otros motores le superaran poco después de su lanzamiento.



Ilustración 23.Doom III.

- **DarkBasic.** Se trata de un motor para desarrolladores amateur, en contraposición con los anteriores que se especializan en proyectos más profesionales. Se debe destacar, por ser uno de los antecesores de *Blender*.
- **MassTech.** Nace en 2004 y se convierte en un duro competidor tanto para *Id Tech* como para *Unreal*. Dispone de las librerías gráficas más importantes hasta la fecha como son *DirectX9*, *DirectX10* y *OpenGL*. Y, por supuesto, se trata de un motor de juego con capacidad de diseño multiplataforma. Es uno de los mejores motores que podemos encontrar hoy en día en el mercado.
- **CryENGINE.** Considerado dentro de los mejores, nació en 2004 desarrollado por la empresa turca *Crytek*[26]. Es gráficamente exquisito por lo que sus proyectos son de una calidad espectacular. Se diseñó con el fin de demostrar la capacidad de la nueva tarjeta *Nvidia Geforce 3*, mediante una demo llamada *X-Isle*. Más tarde se extendió a las videoconsolas, gracias a la posibilidad del diseño multiplataforma.
- **Source.** Toda la evolución tecnológica vista en los motores anteriores desemboca en el que según los expertos es el mejor motor, hasta el momento. Convirtiéndose en el estándar a partir del cual se desarrollarían todos los

proyectos posteriores. La implementación de la lógica de los videojuegos es muy sencilla y dispone de mucha documentación lo que facilita el aprendizaje de la herramienta. Se pueden realizar grandes productos a un coste mucho menor que con los otros motores disponibles. A todo ello, se le une una política de distribución envidiable, por lo que grandes desarrolladores lo eligieron para la realización de sus proyectos, obteniendo un gran reconocimiento y lanzando al mercado un producto de gran rentabilidad.



Ilustración 24. Left 4 Dead.

- **RAGE.** Nacido en 2006 de la colaboración de *RAGE Technology Group*[27] y *Rockstar*. Este motor combina un *framework* de renderización, un motor para la física de los objetos, un motor de audio, librerías de red, un motor de animación y un lenguaje scripting entre otras cosas. Como se puede apreciar es muy completo a la vez que intuitivo en el manejo. Hay que destacar que el motor de animación no es de creación propia sino que integraron uno de los mejores que existían en el momento, *Euphoria*[28]. Lo mismo ocurrió con el motor de la física, recurriendo a *Bullet*. Este motor de juego mezcla una gran calidad gráfica que introduce al usuario a un mundo casi real y, además, una calidad física de los objetos y del movimiento espectacular.



Ilustración 25. Grand Theft Auto IV.

A partir de este momento, las empresas realizan un mantenimiento de los motores que se ajustan al estándar marcado por *Source*, añadiendo nueva mejoras. Sin embargo se está viviendo una desaceleración del mercado de los motores de juego para PC debido al nacimiento de las videoconsolas de última generación, lo que desemboca en que los desarrolladores de nuevos productos estén centrados en el diseño en estas plataformas.

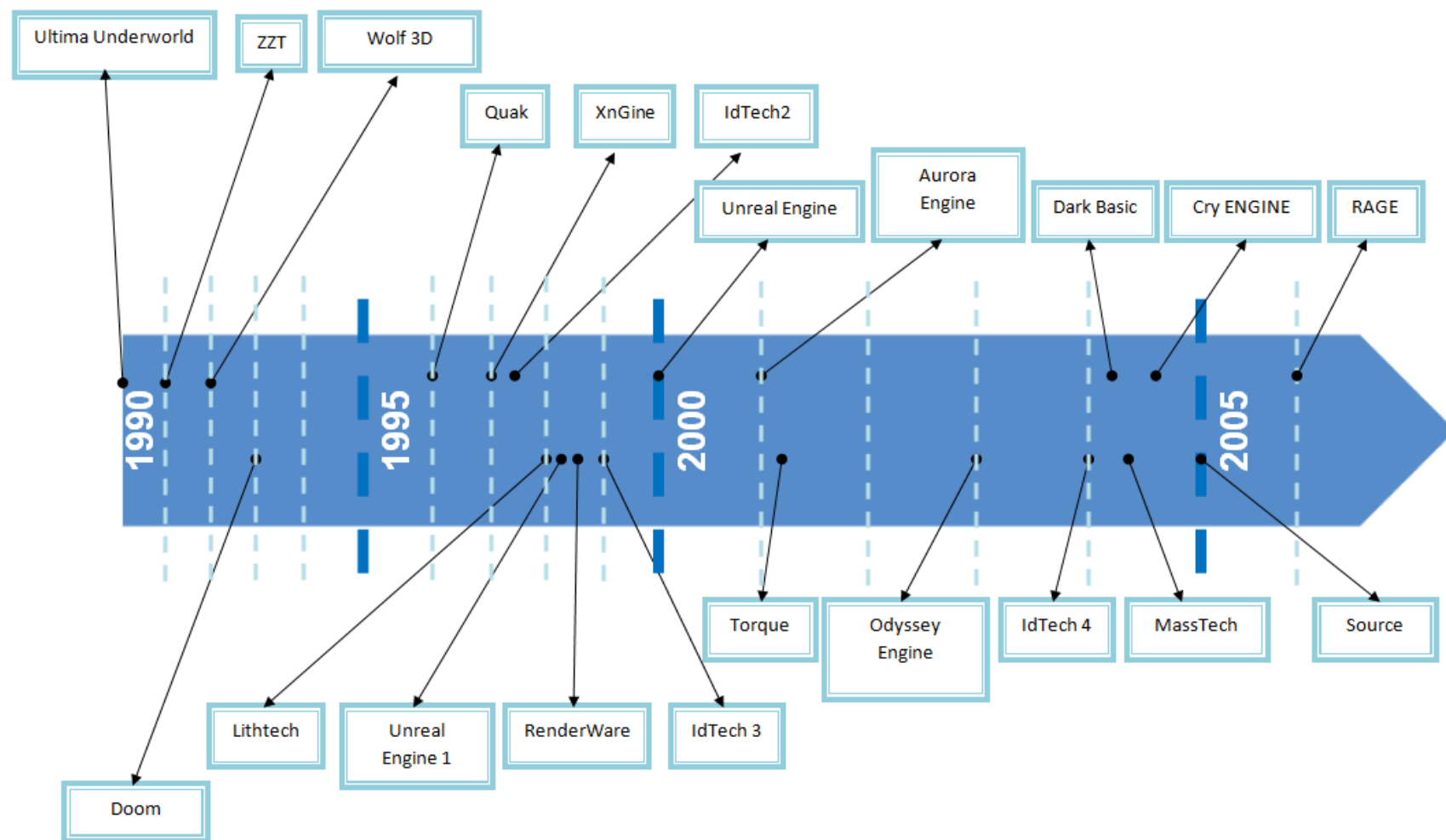


Ilustración 26. Eje cronológico de los motores de juegos vistos hasta el momento.

2.2.2. Motores de juego en la actualidad.

Se puede decir que actualmente los motores de juego se diferencian principalmente por un único concepto, el grado de conocimiento de programación que se necesita para su utilización.

- **Roll-your-own.** Algunas empresas independientes continúan con el concepto de desarrollar su propio motor para la creación del videojuego. Para ello utilizan *APIs* disponibles, como *DirectX* u *OpenGL*, y librerías que podemos obtener en el mercado para optimizar el desarrollo. Como se puede intuir, esto proporciona mayor flexibilidad y libertad de desarrollo, pero supone un mayor tiempo de creación. Por tanto, estos motores son poco atractivos para los desarrolladores de juegos.
- **Mostly-ready.** Estos motores incluyen las herramientas de renderizado, la interfaz gráfica de usuario y la física, entre otras muchas cosas. La mayoría incluyen *OGRE*[29] y *Genesis3D*[30] que son motores de renderizado 3D en código abierto. Dentro de esta categoría están la mayor parte de los motores disponibles, como por ejemplo *Torque* o *Unreal*. Estos motores requieren que el desarrollador tenga que programar determinados aspectos de su juego, mediante secuencias de comandos o incluso realizar una programación a bajo nivel. Tienen más limitaciones que los anteriores pero son óptimos y proporcionan un mayor rendimiento a un menor coste.
- **Point-and-click.** Incluyen una serie de herramientas que le permiten a cualquier usuario crear un juego de una manera muy fácil e intuitiva. Se evita por todos los medios que el usuario tenga que programar, por tanto, son motores que disponen de una gran interfaz gráfica, que llega a ser bastante compleja de manejar. Son muy limitados y, sin embargo, es posible realizar juegos muy creativos. Además, permiten trabajar de forma muy rápida.

Vistos los tres enfoques, podemos decir que *Blender* se establecería en el último de ellos. Más adelante se explicará esta herramienta de manera más detallada.

En la siguiente tabla se realiza una comparativa entre los tres, destacando algunas características importantes en el desarrollo de videojuegos. Esto no quiere decir que uno sea mejor que otro, simplemente debemos elegir el que más se adapte a nuestra situación y al resultado que se espera obtener.

	Roll-your-own	Mostly-ready	Point-and-click
Flexibilidad y libertad de desarrollo	◆◆◆	◆◆	◆
Tiempo de creación	◆◆◆	◆◆	◆
Nivel de programación exigido	◆◆◆	◆◆	◆
Usabilidad	◆	◆◆	◆◆◆
Resultado obtenido	◆◆◆	◆◆	◆
Rentabilidad	◆	◆◆◆	◆◆
A quién van dirigidos	Desarrolladores muy expertos, con un alto nivel de exigencia. Quiere obtener mejores resultados sin importarle el coste asociado. Industria independiente.	Desarrolladores expertos, que buscan un equilibrio entre resultados y costes. Profesionales.	Desarrolladores poco expertos, que no se embarcan en grandes proyectos, pero que consiguen resultados más que aceptables. Industria amateur.

Tabla 1. Clasificación tipos de motores de juego.

Aunque el motor de juego elegido para la realización del proyecto ha sido *Blender*, en el mercado disponemos de otros que nos ofrecen mejores resultados y que son los que en su mayoría utilizan los que se dedican profesionalmente a la creación de videojuegos.

A continuación, vamos a citar algunos de los más importantes y a describir algunas de sus características para realizar una panorámica de las diversas opciones que un desarrollador tiene a su alcance.

- **CryEngine 3D** [26].

Se trata de un motor gráfico que dispone de un potente renderizado tanto de escenarios interiores como exteriores, consiguiendo lo que se denomina foto-realismo. Es una aplicación muy escalable ofreciendo al desarrollador diseñar sus videojuegos con independencia de las plataformas en la que se vaya a comercializar.

Ofrece técnicas de animación avanzada para realizar personajes con gráficos realmente asombrosos para el diseño de juegos multiplataforma. Su motor para la generación de la física se puede aplicar a cualquier objeto que

forme parte del juego, sin la necesidad de integrar in *middleware* externo. También, dispone de un sistema de edición de audio profesional para la introducción de sonidos de calidad real en un videojuego. Por último, destacar su herramienta *sandbox*, ofreciendo a los usuarios un control total de sus proyectos multiplataforma en tiempo real.

- **Torque 3D**[31].

Es un motor de juego que dispone de un conjunto de herramientas para producir simulaciones y videojuegos multiplataforma. Con este motor podemos realizar cambios y ver el resultado prácticamente en tiempo real, además es compatible con algunas de las herramientas de modelado 3D más importantes del mercado, como son *3D Studio* o *Blender*, entre otras.

En cuanto al renderizado, posee una de las tecnologías líderes en esta técnica, incluyendo iluminación por píxel dinámico, norma u oclusión por paralelismo, además de un sistema para el sol, las estrellas, las nubes y las explosiones.

Sin embargo, la herramienta de simulación de la física, el editor de sonido y otras herramientas no se incluyen en el producto.

- **Source Engine**[32].

Está considerado como uno de los motores de juego más potentes del mercado. Ofrece todas las herramientas conocidas hasta ahora en cuanto a la creación de videojuegos, sistema de renderizado eficiente, posibilidad de modelar y animar objetos 3D, diseño de escenarios mediante mallas, integración de un potente simulador de la física, implementación de mecanismos de juego con un sofisticado sistema de inteligencia artificial y editor de texto para implementación propia y editor de audio, entre otras muchas cosas.

Es muy complicado determinar cuál es el mejor motor de juego en la actualidad, debido a que cada uno nos ofrece característica que son subjetivas en su análisis. Sin embargo, en muchas publicaciones y artículos en los que se ha hablado sobre los motores de juego, siempre han sido citados los explicados anteriormente.

2.3. BLENDER ENGINE.

Blender es un programa informático dedicado al modelado, animación y creación de gráficos tridimensionales. Por establecer una analogía, podemos decir que *Blender* es el *Fotoshop* del modelado 3D.

Nació en 1993 integrando una serie de herramientas para la creación de contenidos en 2D y en 3D. En 2002 la compañía quebró y los acreedores decidieron ofrecerlo como un producto de código abierto y gratuito disponible bajo la licencia *GNU GPL*. En ese momento, los desarrolladores crearon *Blender Foundation*, una asociación sin ánimo de lucro para recoger donaciones con el fin de ofrecer un mantenimiento del producto. Esta decisión dio sus frutos y hoy en día es una de las aplicaciones *Open Source* más populares y utilizadas del mundo.

Se trata de un conjunto de herramientas que permiten la creación y reproducción lineal y en tiempo real de contenidos 3D. Con ellas podemos realizar las siguientes funciones: modelado, renderizado, texturizado, animación, post-producción de audio y video, además de iluminación y creación de juegos. Esto último gracias al motor de juegos que incluye el programa *Blender Game Engine*, sumado al hecho de poder operar en multitud de plataformas, exportar en multitud de formatos y con un tamaño de descarga de archivos muy reducido.

Las principales características son:

- **Interfaz gráfica de usuario, GUI.** Se trata de un entorno de trabajo con un sistema de ventanas sin solapamiento, que pueden ser totalmente configuradas de forma independiente al gusto del usuario. Además, ofrece un editor de texto utilizado principalmente para la creación de scripts en *Python*.
- **Modelado.** Se puede realizar un modelado profesional con las herramientas que proporciona. Estas se componen de primitivas geométricas como son mallas poligonales, objetos *empty*, *NURBS* y *metaballs*, con los que empezar a modelar y herramientas más especializadas para realizar un acabado limpio.
- **Rigging.** Este sistema permite crear esqueletos de forma sencilla de tal manera que agregándole una malla, pueda controlarse el conjunto de forma automática, para aplicarle movimientos que parezcan fluidos y reales.
- **Animación.** Herramienta para asociarle movimientos a los objetos, por tanto a partir de esqueletos como de *frames*. Mediante *IK chaining* podemos asociar los movimientos de cambios en el desplazamiento, tamaño o rotación de un objeto entre dos *frames* mediante un indicador que los agrupa, facilitando el manejo del

movimiento que sufrirá el objeto en cuestión. Además, nos permite crear nuevos objetos a partir de otros y mezclar dichos movimientos con efectos sonoros, pudiendo reproducirlos, mezclarlos y editarlos.

- **Mapeado de texturas UV.** Nos permite texturizar los objetos con la imagen que queramos, pudiendo ajustarla a nuestro gusto. Además, nos proporciona distintas operaciones como dividir o replicar la textura.
- **Renderizado.** Se basa en un sistema de renderizado interno, aunque podríamos integrar otros que incluyan un potente trazado de rayos como *kerkythea* o *YafRay*. *Blender* realiza el renderizado por secciones, incluye capas y una caché de renderizado. También puede proporcionar soporte para efectos como destellos de luz, oclusión ambiental o desenfoque.
- **Shading.** Dispone de una herramienta para agregarle un material y su sombra correspondiente a un objeto de manera muy sencilla. Distingue dos tipos de sombreado; difuso y especular. Dispone también de un editor de nodos para crear efectos visuales en el material asociado al objeto. Es muy útil en el caso de querer representar materiales como la madera o el metal, o incluso la representación de fluidos.
- **Física y partículas.** Se utiliza para simular cabello o humo de una manera muy realista. Sin embargo implica un alto coste computacional. Se le asigna un determinado número de partículas a un objeto, pudiendo controlar su movimiento, su texturizado y su color. Mediante la física, podemos cambiar la percepción del objeto de tal manera que podamos simular un cuerpo rígido, como una bola de metal, un cuerpo suave, como una pelota de goma o un fluido. De esta manera se cambian las propiedades físicas de los objetos para que visualmente se perciban como en la realidad. También incluye detección de colisiones, muy útil a la hora de diseñar un videojuego.
- **Creación de contenidos 3D en tiempo real.** Proporciona una herramienta para la inserción de la lógica de un videojuego sin necesidad de tener que implementar un código propio, y además de manera muy sencilla e intuitiva. Permite la introducción de scripts *Python* creados por el desarrollador para añadir funcionalidades que no se puedan realizar en esta ventana, gracias a la librería *GameLogic* integrada en el programa.
- **Gestión de archivos.** *Blender* guarda el modelo creado en un archivo *.blend*, incluyendo la física, las texturas, los materiales y la iluminación. Estos ficheros proporcionan compatibilidad hacia delante y hacia atrás, admiten compresión y

encriptación, y pueden servir como librerías en otros archivos del mismo tipo. Además, los proyectos en *Blender* pueden ser exportados a multitud de formatos, entre los que está *3DStudio*, *DirectX* y *Quake 3*.

- **Soporte multiplataforma.** *Blender* está disponible en multitud de plataformas y mantiene su estructura de ventanas en todas ellas:
 - Windows 2000, XP, Vista
 - Mac OS X (PPC and Intel)
 - Linux (i386)
 - Linux (PPC)
 - FreeBSD 5.4 (i386)
 - SGI Irix 6.5
 - Sun Solaris 2.8 (sparc)

3. ANÁLISIS.

La fase de análisis es esencial en cualquier proyecto comercial. Su finalidad es estudiar a fondo los requisitos del cliente con el fin de escoger la mejor solución posible con las tecnologías idóneas para su desarrollo, y además, optimizar los recursos utilizados para ello.

En primer lugar, se va a explicar brevemente la propuesta inicial proporcionada por el cliente, para en el siguiente punto, describir los requisitos que ha establecido. Estos requisitos son los que marcan las pautas para el posterior diseño del videojuego.

Por último, una vez analizados los requisitos, se buscan las mejores soluciones, se le muestra al cliente la propuesta junto con un presupuesto y una estimación de la duración del desarrollo, para que este decida si se debe o no continuar con el proyecto. Si finalmente decide contar con nuestros servicios, se comienza con el diseño.

El diseño de un videojuego se tiene que establecer en varias fases. Al comienzo de la primera fase el cliente muestra sus requisitos y los desarrolladores comienzan el diseño a partir de ellos. Al final de esta fase, se celebra una reunión para mostrar el estado de proyecto, se podría decir que se trata de la versión *alfa* del videojuego, en ella se discuten las soluciones de los requisitos por si hay que hacer modificaciones. Además, el cliente puede volver a definir requisitos nuevos, que han surgido a raíz de la realización de la primera versión. Este proceso se repetiría, para la versión *beta* y para la versión final, antes de la comercialización del videojuego.

Por tanto, lo que veremos a continuación será el análisis del proyecto desde el principio y hasta el final de la primera fase, donde obtendremos la resolución de los requisitos iniciales y definiremos su solución, para luego, en el siguiente punto, desarrollarlos y obtener la versión *alfa* del videojuego, para así realizar una fase de pruebas en la que veremos los errores que hay que solucionar y los nuevos requisitos del cliente para la fase *beta*.

3.1 PROPUESTA INICIAL.

Nos vamos a centrar en el siguiente caso práctico, una empresa solicita nuestros servicios para desarrollar un videojuego de plataformas. Quieren un juego sencillo que

consiga llamar la atención de un público que lo usará como sistema de entretenimiento durante cortos espacios de tiempo.

El cliente establece que la tecnología utilizada sea en 3D, permitiéndonos estudiar la mejor opción.

La propuesta inicial estudiada es la realización de un juego donde el personaje es una bolita que tiene que ir por un mundo de plataformas consiguiendo superar los retos que se encuentran por el camino. Para superar los retos, el usuario tendrá que tener habilidad, memoria y visión espacial.

3.2 ANÁLISIS DE LAS HERRAMIENTAS A UTILIZAR.

Conociendo ya la propuesta inicial del cliente, antes de iniciar el desarrollo del proyecto como tal, debemos analizar todas las herramientas disponibles en el mercado, tanto para el modelado como para la implementación, para luego tomar la decisión correcta en cuanto a la herramienta o herramientas que vamos a utilizar.

Para conocer las diferencias existentes tanto entre las herramientas de modelado 3D como entre los motores de juego disponibles en el mercado, debemos analizar las siguientes características:

- Open Source. Si la herramienta tiene licencia abierta o lo que es lo mismo, si se trata de una aplicación gratuita o no.
- Curva de aprendizaje. Si se trata de una herramienta fácil de utilizar o si necesita de mucho tiempo de aprendizaje para poder llegar al resultado que queremos.
- Resultado. Cuál es el grado de perfeccionamiento obtenido con la aplicación, si los modelos realizados con ellas son realistas.
- Profesional. Si la herramienta es usada por desarrolladores de videojuegos profesionales.

Mediante la siguiente tabla, vamos a realizar la comparativa entre todas las herramientas descritas en el apartado “Herramientas de modelado 3D en la actualidad” para luego determinar la que vamos a utilizar.

Herramienta	Open Source	Curva de aprendizaje	Resultado	Uso profesional
AutoDesk Maya	NO	Alta	Bueno	SI
SoftImage	NO	Alta	Bueno	SI
3D Studio	NO	Media-Baja	Bueno	SI
LightWave	NO	Media-Baja	Bueno	SI
Blender	SI	Media	Bueno	NO
Cinema 4D	NO	Media-Baja	Muy bueno	NO
Houdini	NO	Alta	Muy bueno	SI
Rhinoceros	NO	Baja	Bueno	NO
Cheetah 3D	NO	Baja	Bueno	NO
ZBrush	NO	Muy alta	Muy bueno	NO

Tabla 2. Herramientas de modelado.

A continuación se muestra la misma tabla comparando los motores de juego existentes en la actualidad.

Herramienta	Open Source	Curva de aprendizaje	Resultado	Uso profesional
CryEngine 3D	NO	Alta	Muy bueno	SI
Torque 3D	NO	Alta	Muy bueno	SI
Source Engine	NO	Alta	Bueno	SI
Blender	SI	Media	Bueno	NO

Tabla 3. Motores de juego.

Analizando las dos tablas anteriores, y teniendo en cuenta que la experiencia previa en el diseño de videojuegos era prácticamente nula, se ha optado por elegir una herramienta con la que se pudiera obtener un buen resultado y que no tuviera asociada una curva de aprendizaje excesivamente alta.

Otro aspecto a tener en cuenta es que, como ya hemos estado viendo a lo largo de la memoria, es necesario el uso de dos herramientas para el desarrollo del videojuego, una herramienta de modelado 3D y un motor de juego. Muchas de las aplicaciones que hemos visto, tanto de modelado como de animación, incluían herramientas contenidas en ambas. Sin embargo el factor de decisión es el coste de las herramientas.

Por todo esto, la herramienta utilizada es *Blender*, ya que como se ha explicado detalladamente en el punto 3.6, se trata de una buena herramienta para el diseño e implementación en 3D para personas que se quieren iniciar en este campo y que además, es una aplicación *Open Source*. Y que, por supuesto, nos ofrece el resultado esperado para los objetivos que nos hemos fijado.

Con la propuesta inicial ya definida el cliente nos describe sus requisitos, para que procedamos a su estudio con el fin de llegar a la mejor solución.

3.3 ANÁLISIS DE LOS REQUISITOS.

El cliente describe una serie de requisitos que para él son imprescindibles en el videojuego. Estos son requisitos muy generales y los desarrolladores deben desglosarlos en requisitos más específicos, para abordar temas importantes que el cliente no tiene en cuenta, pero que influyen en sus exigencias, y además para poder realizar un mejor estudio.

Los requisitos establecidos por el cliente son:

- El videojuego tiene que ser desarrollado mediante tecnología 3D.
- Tiene que tener un archivo ejecutable para que el usuario pueda arrancar el videojuego de una manera sencilla.
- Debe tener efectos sonoros.
- Tiene que disponer de menús de navegación para que tenga una buena usabilidad. Y estos menús deben estar escritos en inglés.
- Los controles para el manejo del juego deben ser comunes y que los usuarios tengan a su disposición.
- El juego debe ser compatible para el sistema operativo Windows.

Una vez vistos los requisitos del cliente, vamos a proceder a su desglose y a proponer las distintas soluciones. Para cada uno de los requisitos tenemos una tabla en la que podemos ver:

- El desglose que se cree necesario dentro de cada requisito estipulado por el cliente.
- La importancia del requisito:
 - Obligatorio (R, *Required*), necesario para que se cumplan la especificaciones del cliente.

- Aconsejable (A, *Advisable*), no es necesario para cumplir las especificaciones en esta primera fase, pero serán requisitos obligatorios en las próximas.
 - Opcional (O, *Optional*), no está dentro de las especificaciones del cliente, sin embargo ayuda al resultado final del proyecto.
- La descripción del requisito como tal.
 - La solución tecnológica propuesta en el diseño.

Requisito 1: Tecnología 3D.			
Desglose	Importancia	Descripción	Solución
Modelado 3D	R	El juego tiene que estar modelado en 3D.	El modelado se realizará con la herramienta <i>Blender</i> , y el motor de juego será <i>Blender Game Engine</i> , ya que nos da grandes resultados y además se trata de una aplicación <i>open source</i> .
Motor de juego	R	La animación del juego debe realizarse mediante un motor de juego.	

Tabla 4. R1 Tecnología 3D.

Requisito 2: Plataforma en la que se ejecutará.			
Desglose	Importancia	Descripción	Solución
Sistema operativo Windows	R	El sistema operativo será Windows Vista.	El archivo ejecutable deberá ser un .exe, para que el juego se pueda ejecutar en Windows Vista.
Compatibilidad hacia atrás	O	El juego deberá ejecutarse en las versiones anteriores de Windows.	La compatibilidad hacia atrás será posible con la solución anterior, sin embargo se deberán tener en cuenta los requerimientos hardware necesarios para su ejecución, como por ejemplo, disponer de una tarjeta gráfica potente para el renderizado de los gráficos.
Compatibilidad con otros sistemas operativos.	O	El juego deberá ser compatible en otros sistemas operativos	Se deberá generar un ejecutable para los distintos sistemas operativos.
Sistema multiplataforma	O	El juego deberá migrar hacia otras plataformas.	Para la migración hacia otras plataformas habrá que estudiar los diferentes sistemas operativos de cada una de ellas.

Tabla 5. R2 Plataforma en la que se ejecutará.

Requisito 3: Sonido.			
Desglose	Importancia	Descripción	Solución
Efectos sonoros	R	El juego tiene que tener efectos sonoros.	Los efectos sonoros serán archivos .mp3 y se reproducirán en función de la implementación

			del juego.
Música de fondo	A	El juego debe tener música durante toda la ejecución.	La música de fondo se reproducirá del mismo modo al requisito anterior.

Tabla 6. R3 Sonido.

Requisito 4: Controles.

Desglose	Importancia	Descripción	Solución
Controles básicos	R	El juego debe controlarse con un periférico sencillo y que un usuario pueda disponer fácilmente.	El usuario podrá controlar a su personaje desde las flechas que se encuentran en el teclado del ordenador.

Tabla 7. R4 Controles.

Requisito 5: Interfaz de usuario.

Desglose	Importancia	Descripción	Solución
Menús	R	El juego debe tener menús para que el usuario pueda interactuar con el juego de una manera intuitiva.	El juego dispondrá de un menú sencillo de fácil manejo con diversas opciones.
Ayuda	R	El juego debe tener una ayuda para que el usuario conozca a priori un poco sobre la mecánica del juego.	Se accederá a la ayuda a través del menú principal y explicará conceptos como los controles o algunos objetos que le ayudarán para completar el juego.
Elección de la pantalla	R	El usuario debe poder elegir la pantalla en la que quiere jugar	El usuario podrá acceder a un menú desde una de las opciones del menú principal
Salir del juego	R	El usuario debe tener opción a abandonar el juego cuando él desee.	Tendrá una opción desde el menú y además, tanto si supera el nivel, como si no, también dispondrá de esa opción.
Menú control sonido	A	El usuario debe tener un menú para controlar el volumen del sonido o su desactivación si lo quisiera.	Accederá a un menú de control de sonido desde una de las opciones del menú principal.
Idioma: Inglés	O	Los menús de la GUI deben estar en inglés.	El idioma de los menús estarán escrito en inglés.

Tabla 8. R5 Interfaz de usuario.

Requisito 6: Ejecutable.

Desglose	Importancia	Descripción	Solución
Archivo ejecutable	R	El juego debe ser ejecutado a partir de	El usuario dispondrá de un archivo ejecutable, a partir de

		un único archivo ejecutable.	cual se ejecutará todo el juego.
Instalador	O	El juego debe incorporar un instalador para instalar todos los archivos necesarios de manera automática.	En caso de necesitar archivos .dll u otros para su ejecución, se empaquetarán todos los archivos y dispondrá de un instalador.
Desinstalador	O	El juego debe incorporar un archivo para desinstalar todos los archivos instalados con la versión empaquetada.	En el paquete de la versión final se introducirá un archivo desinstalador.

Tabla 9. R6 Ejecutable.

Conociendo ya los requisitos estipulados por el cliente, y una vez estudiados, por separado cada uno de ellos, mediante un desglose para una comprensión mayor, podemos finalizar el estudio, exponiendo los requisitos software.

Los requisitos software son los requisitos definitivos, con los cuales los desarrolladores van a proceder al diseño de su aplicación, y que surgen a raíz del estudio de los del cliente.

Por tanto, los requisitos software necesarios para el desarrollo de la fase *beta* de este videojuego vienen explicados en la siguiente tabla.

Tipo de requisito software	Nombre del requisito	Importancia
Requisitos funcionales	Instalador del juego	O
	Ejecutable del juego	R
	Desinstalador del juego	O
	Menú principal	R
	Menú de elección de pantalla	R
	Controles	R
	Pantalla de ganador	R
	Pantalla de perdedor	R
	Salir del juego	R
	Menú control de audio	A
Requisitos de rendimiento	Tiempo de carga del juego	A
	Tiempo de carga de las pantallas	A

	Información de carga	O
Requisitos de la GUI	Buena resolución de los menús	R
	Buena resolución de los objetos 3D	R
	Idioma: Inglés	O
Requisitos de recursos	Entorno de desarrollo	A
	Entorno de producción	A

Tabla 10. Requisitos software.

El requisito más importante para los desarrolladores es el requisito que especifica los recursos que se van a utilizar. A partir de estos, los desarrolladores tienen que determinar la tecnología hardware que van a utilizar para su proyecto. Y esto, a su vez, afectará a los requerimientos hardware y software que se les impondrá a los usuarios.

Requisito para el desarrollador (entorno de desarrollo)	
Descripción	Características mínimas del hardware
Para el entorno de desarrollo es necesario un ordenador que disponga de una memoria RAM alta y una buena tarjeta gráfica.	Sistema operativo: Windows Vista
	Entorno de desarrollo adicional: Python
	Memoria RAM: 4G
	Capacidad mínima de disco duro: 30 G
	Tarjeta gráfica mínima: 512MB

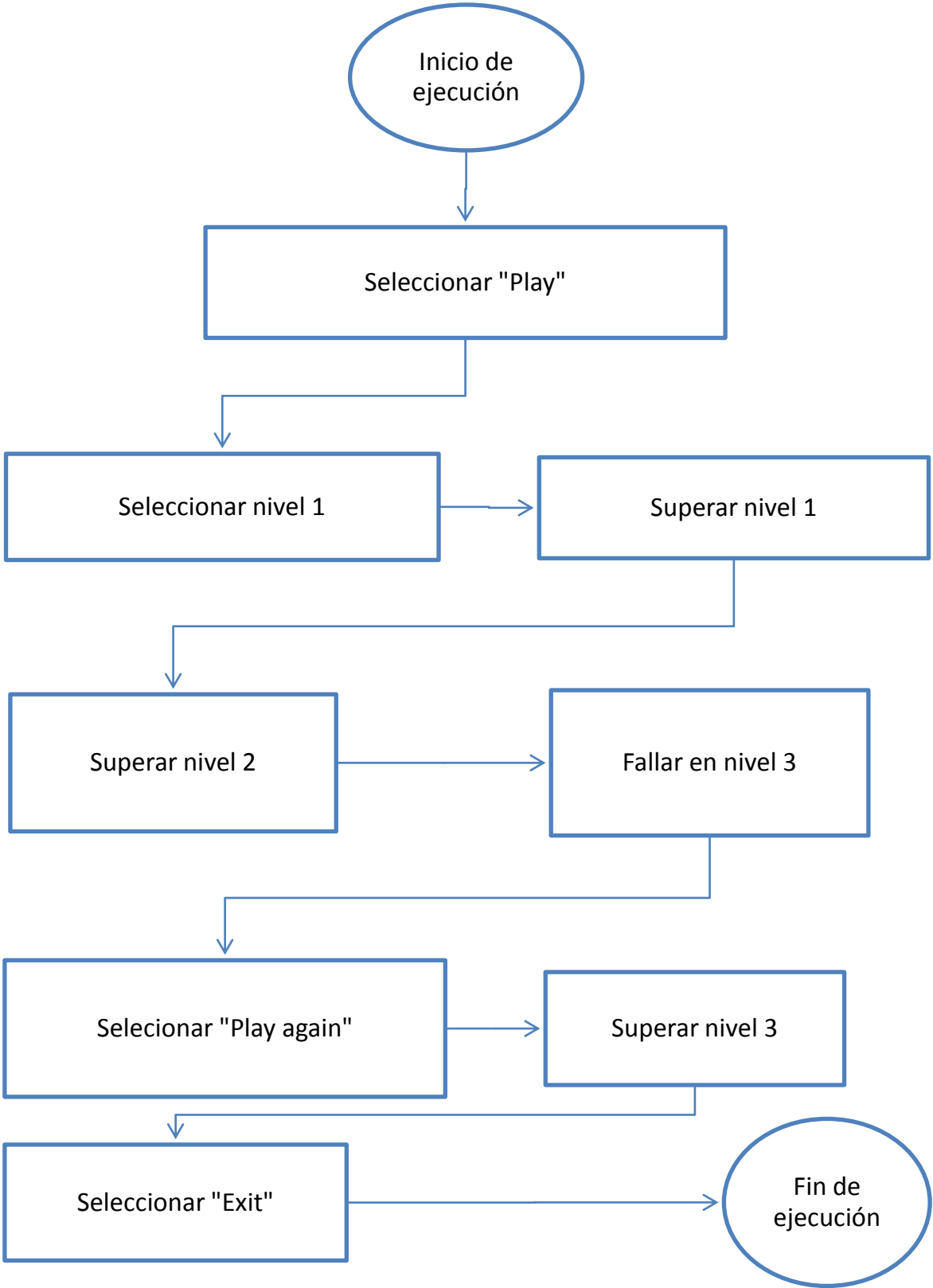
Tabla 11. Requisitos del entorno de desarrollo.

Requisito para el usuario (entorno de producción)	
Descripción	Características mínimas del hardware
El usuario debe disponer de un ordenador con las característica que se enumeran a continuación, además de unos altavoces para la reproducción de los sonidos	Sistema operativo: Windows Vista o superior.
	Memoria RAM: 2G
	Capacidad mínima de disco duro: 400MB
	Tarjeta gráfica mínima: 256MB

Tabla 12. Requisitos del entorno de producción.

3.4 CASO DE USO

A continuación, vamos a mostrar un posible caso de uso de la aplicación.



3.5 DIAGRAMA DE EJECUCIÓN.

Una vez explicados los requisitos del cliente y las soluciones propuestas, se va a mostrar en el siguiente diagrama cómo es el flujo de ejecución, para conocer mejor el comportamiento del juego a través de su diagrama de estado.

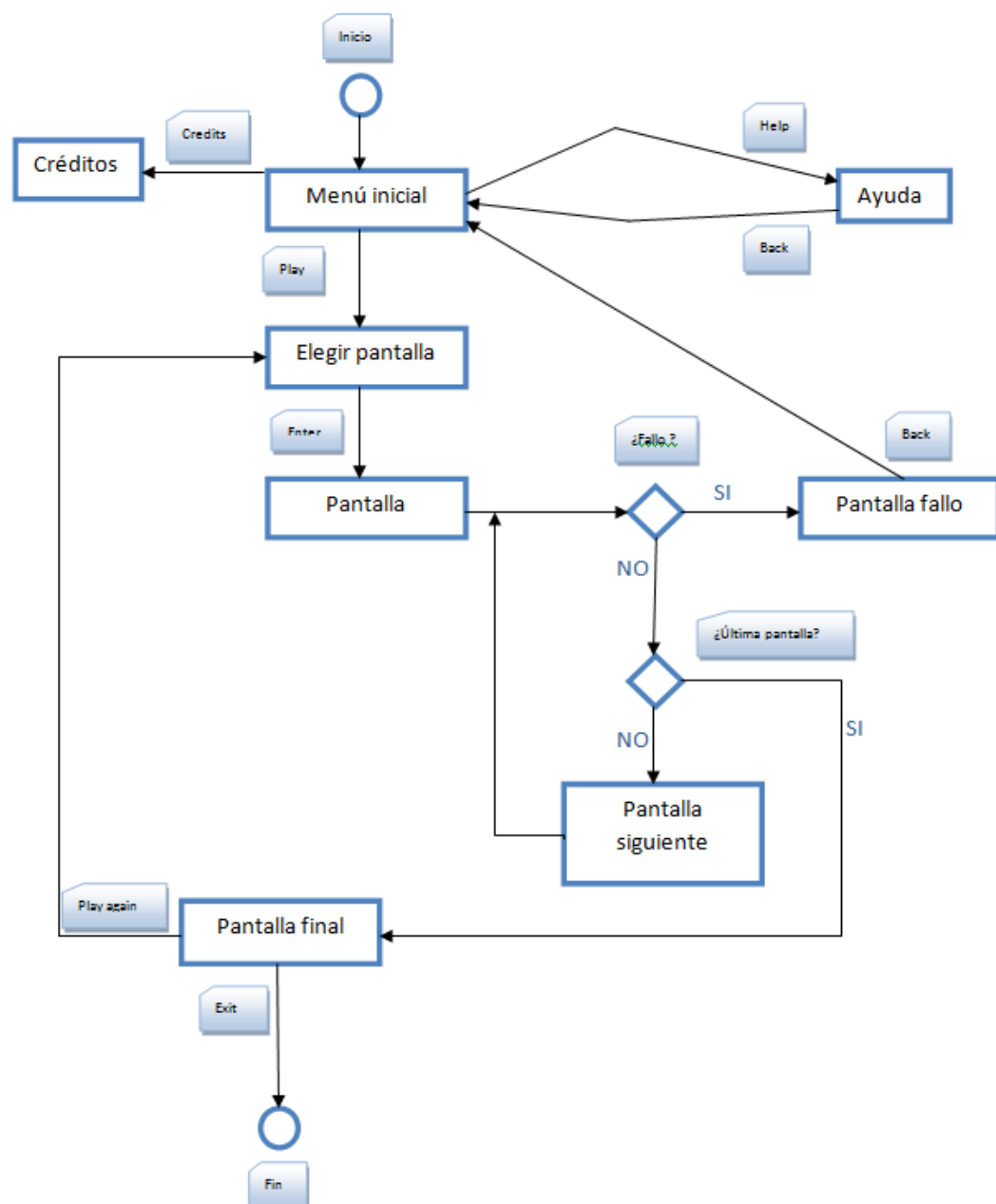


Ilustración 27. Diagrama de ejecución.

3.6 ESTIMACIÓN Y PRESUPUESTO INICIAL

Cuando ya se ha realizado un estudio exhaustivo de los requisitos, se han valorado todas las opciones y se ha llegado a una solución final, es necesario hacer una estimación para saber de manera aproximada cuánto tiempo se necesita para su realización.

Para este proyecto, se ha hecho una estimación del tiempo necesario para la realización de la versión *beta* de un videojuego, a partir de los requisitos especificados por un cliente ficticio. Esta estimación nos orienta sobre el tiempo aproximado de realización del proyecto y nos permite realizar un presupuesto inicial.

La estimación se ha realizado en base a los siguientes puntos:

- La documentación de la herramienta *Blender* y el aprendizaje necesario para realizar el proyecto.
- El diseño del videojuego.
- La fase de *testing* con su análisis correspondiente,
- La documentación y escritura de la memoria.

Para cada uno de ellos hemos estimado el tiempo que inicialmente se cree necesario para su desarrollo. Luego, mediante la herramienta "*GanttProject*", hemos obtenido el diagrama de GANTT⁴ y de PERT⁵ asociado al proyecto.

⁴ Ver ANEXO II. DIAGRAMA DE GANTT INICIAL.

⁵ Ver ANEXO III. DIAGRAMA DE PERT INICIAL.

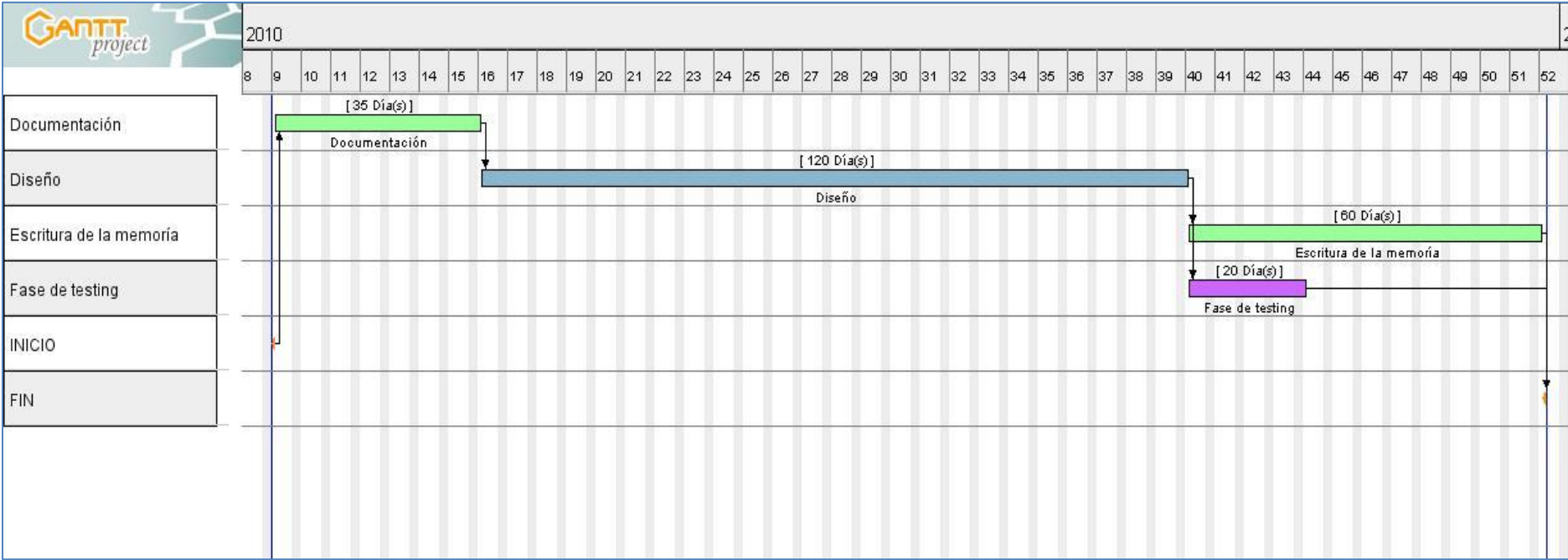


Ilustración 28. Diagrama de GANTT inicial.

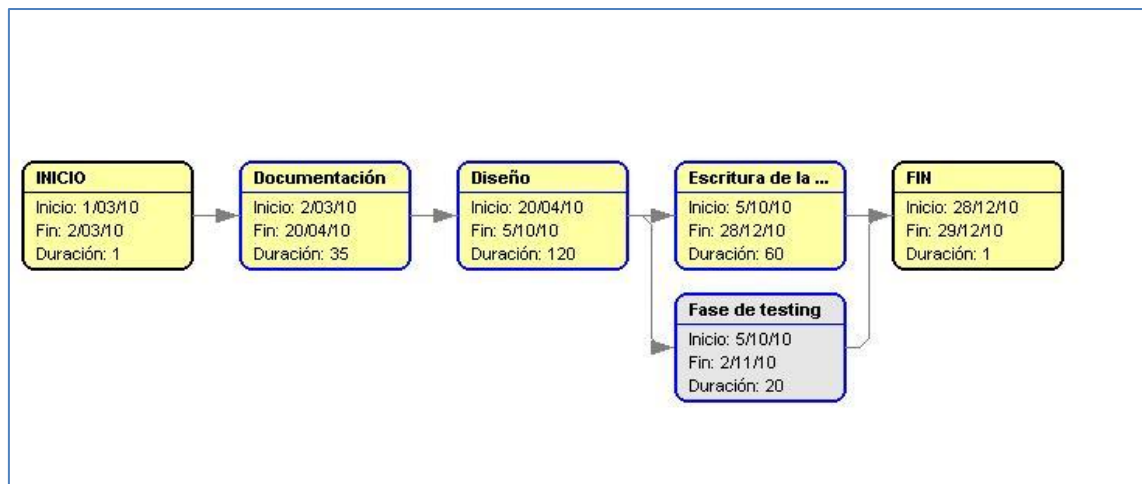



Ilustración 29. Diagrama de PERT inicial.

En la siguiente tabla, se muestra el proyecto desglosado en tareas, cada una con la fecha de inicio y fin estimada.

Tarea	Fecha de inicio	Fecha de fin	Tiempo estimado (días laborales)
Documentación	2/03/10	20/04/10	35
Diseño	20/04/10	5/10/10	120
Escritura de la memoria	5/10/10	28/12/10	60
Fase de testing	5/10/10	2/11/10	20
TOTAL:			235

Tabla 13. Estimación inicial.

Teniendo en cuenta esta estimación inicial podemos realizar su presupuesto asociado. En un caso real, este presupuesto se le mostrará al cliente para que tome la decisión de si somos nosotros los elegidos para la realización del proyecto.

	UNIVERSIDAD CARLOS III DE MADRID Escuela Politécnica Superior
	PRESUPUESTO INICIAL DE PROYECTO
1.- Autor: Ana Román Rodríguez	
2.- Departamento: Informática	

3.- Descripción del Proyecto: Implementación de un videojuego con la herramienta Blender.					
- Título					
- Duración (meses): 2					
Tasa de costes Indirectos: 20%					
4.- Presupuesto total del Proyecto (valores en Euros): 26.000€					
5.- Desglose presupuestario (costes directos):					
PERSONAL					
Apellidos y nombre	Categoría	Dedicación (hombres mes) ^{a)}	Coste hombre mes	Coste (Euro)	Firma de conformidad
				0,00	
Peralta Donante Juan	Ingeniero Sénior	1,79	4.289,54	7.678,28	
Román Rodríguez Ana	Ingeniero	1,79	2.694,39	4.822,96	
				0,00	
				0,00	
	Hombres mes	3,58	Total	12.501,23	
^{a)} 1 Hombre mes = 131,25 horas. Máximo anual de dedicación de 12 hombres mes (1575 horas) Máximo anual para PDI de la Universidad Carlos III de Madrid de 8,8 hombres mes (1.155 horas)					
EQUIPOS					
Descripción	Coste (Euro)	% Uso dedicado proyecto	Dedicación (meses)	Periodo de depreciación	Coste imputable ^{d)}
		100		60	0,00
		100		60	0,00
		100		60	0,00
		100		60	0,00
		100		60	0,00
					0,00
					Total: 0,00
^{d)} Fórmula de cálculo de la Amortización:					
$\frac{A}{B} \times C \times D$ <p> A = nº de meses desde la fecha de facturación en que el equipo es utilizado B = periodo de depreciación (60 meses) C = coste del equipo (sin IVA) D = % del uso que se dedica al proyecto (habitualmente 100%) </p>					

SUBCONTRATACIÓN DE TAREAS		
Descripción	Empresa	Coste imputable
		Total: 0,00
OTROS COSTES DIRECTOS DEL PROYECTO ^{e)}		
Descripción	Empresa	Costes imputable
		Total: 0,00
^{e)} Este capítulo de gastos incluye todos los gastos no contemplados en los conceptos anteriores, por ejemplo: fungible, viajes y dietas, otros,...		
6.- Resumen de costes:		
Presupuesto Costes Totales	Presupuesto Costes Totales	
Personal	12.501	
Amortización	0	
Subcontratación de tareas	0	
Costes de funcionamiento	0	
Costes Indirectos	2.500	
Total	15.001	

4. DESARROLLO DEL PROYECTO

El desarrollo de un videojuego es arduo y costoso. Requiere de un conocimiento sobre la materia bastante amplio, una gran motivación y un nivel de creatividad elevado. Como todo proyecto de desarrollo, se parte de una idea que se va macerando y que culmina con su realización a través de distintas fases:

- Concepción.
- Diseño.
- Planificación.
- Pruebas

Sin embargo, esa idea viene motivada por situaciones y vivencias, que influyen al desarrollador a embarcarse en dicho proyecto.

4.1. INFLUENCIAS.

El mundo de los videojuegos está hoy en día muy presente en nuestra vida. Todos los dispositivos electrónicos que tenemos a nuestro alcance, como teléfonos móviles, *smartphones*, *tablets*, MP4, MP3, PC y ordenadores portátiles, disponen de una oferta de ocio digital asociada a su utilidad principal.

Durante mi formación académica, los videojuegos han sido incluidos en multitud de asignaturas, con el fin de enseñarnos las diferentes tecnologías vigentes y los diversos lenguajes de programación, de una forma amena. Con esto, nos enseñaron la punta de un iceberg, que yo personalmente he querido seguir descubriendo.

Por tanto, una de mis motivaciones era poder contribuir a ello indirectamente mediante el desarrollo de un videojuego propio, y a su vez, aumentar mis conocimientos sobre la materia.

Mi mayor influencia fue el videojuego "*Mercury*", comercializado para la plataforma PSP en el año 2006. Se trataba de un juego de plataformas en 3D, donde el jugador debía llevar a una bola de mercurio a la meta en multitud de niveles.

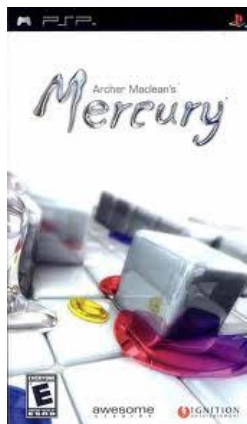


Ilustración 30. Carcasa del juego "Mercury" para PSP.



Ilustración 31. Pantalla "Mercury".



Ilustración 32. Pantalla "Mercury".

4.2. CONCEPCIÓN

Durante esta etapa, es necesario definir ciertos aspectos que son fundamentales para diseñar un videojuego. Estos marcarán el inicio del proyecto y serán primordiales durante las fases que vendrán a continuación:

- **Genero.** Una de las características que describen a un videojuego es su género. Hay multitud de géneros, un ejemplo, son los famosos FPS que se describen en el “Estado del arte” [capítulo 3]. Otro son el género plataformas, donde el jugador controla a un personaje que debe superar diferentes retos dentro de un escenario dado.

Dadas las influencias que he tenido y a los juegos que he disfrutado. Quería que de mi proyecto naciera un juego dentro del género plataforma. Al no ser un videojuego con un desarrollo argumental complejo, tenía que ser más común. Un videojuego de los de antes, pero realizado con la tecnología actual.

- **GamePlay.** Este término describe las experiencias que un jugador tiene al interaccionar con un juego determinado. Esto significa que el desarrollador debe pensar en multitud de situaciones y en especial, los sentimientos que quiere generar y mediante que estímulos.

Mi principal objetivo es que sea un juego divertido, entretenido y fácil de entender, para que cualquier persona sin un conocimiento amplio sobre la mecánica de un juego pueda disfrutar de él.

- **Story Board.** Es un término que se utiliza más en el cine, pero dado que muchos proyectos de videojuegos son casi tan ambiciosos como el séptimo arte, no deberíamos despreciarlo. Se trata de hacer una planificación mediante viñetas de los escenarios del videojuego.

Aunque en mi caso, sea un videojuego muy pequeño, ha sido muy importante realizar bocetos antes de hacer el modelado en 3D de las distintas plataformas. Se podría decir que estos bocetos, son el “story board” de mi videojuego.

4.3. DISEÑO.

El diseño de un videojuego es en gran parte complejo pero a la vez muy satisfactorio. Hay que tener en cuenta los requisitos estipulados por el cliente [apartado 3.2] para empezar el desarrollo. Sin embargo, antes de enfrascarse directamente con la implementación hay que realizar algunas tareas que ayudarán al diseño.

4.3.1. Historia.

Todo videojuego tiene una historia asociada que culmina cuando el jugador consigue el objetivo final. Muchos videojuegos tienen un desarrollo argumental propio de un libro, cargados de personajes, entornos y misiones, que consiguen introducir al jugador en un gran mundo virtual.

En este videojuego, existe un personaje único no humano con forma esférica que debe superar tres niveles, donde encontrará diversos obstáculos que el jugador conseguirá vencer con destreza y habilidad. La dificultad de los niveles irá incrementando según los vaya superando. En el apartado 5.3.4, donde se explica la mecánica del juego, se especificarán los mecanismos que están a disposición del jugador y los obstáculos que deberá vencer para llegar hasta el final del videojuego.

4.3.2. Arte conceptual.

A la hora de modelar un juego en 3D, hay que realizar previamente un trabajo artístico. Es imprescindible plasmar en papel la idea que el desarrollador tiene en mente., por eso, se realizan bocetos de cada uno de los diferentes niveles y de la interfaz gráfica de usuario del videojuego, para que a la hora de empezar el modelado se tenga una idea preconcebida de lo que se desea diseñar. El aspecto visual de cada escenario, así como de la interfaz, es muy importante, el uso de los colores, las luces y las sombras, hace que aumente el realismo.

Para el diseño de los niveles, se partió de un dibujo previo hecho a mano de cada uno de los niveles. Se trata de un guión gráfico, a partir del cual se comienza a diseñar el escenario y que sirve de motor creativo para la introducción de nuevos objetos y la eliminación de otros que a la hora de modelar son, prácticamente, imposibles. Es lo que se denomina “Story Board” del videojuego y viene ilustrado en las ilustraciones 30, 31, 32 y 33.

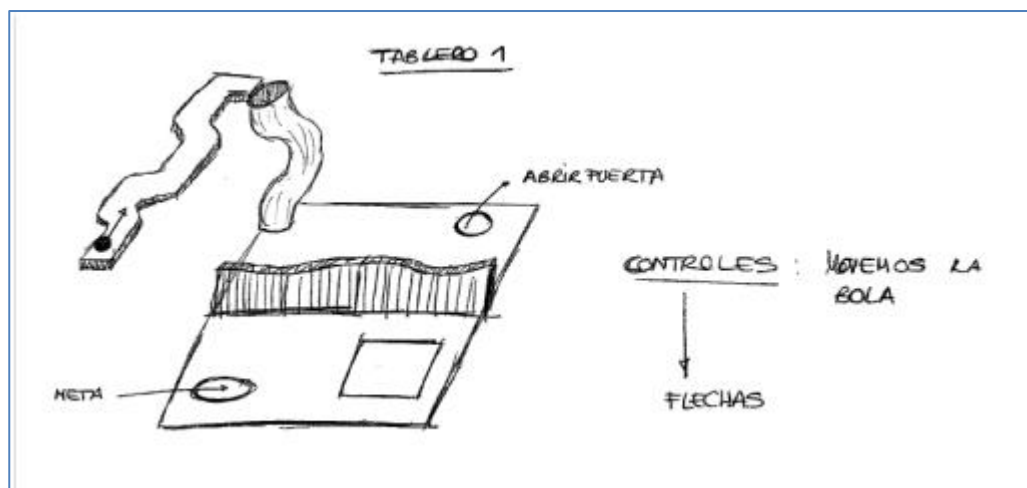


Ilustración 33. Story Board Tablero 1.

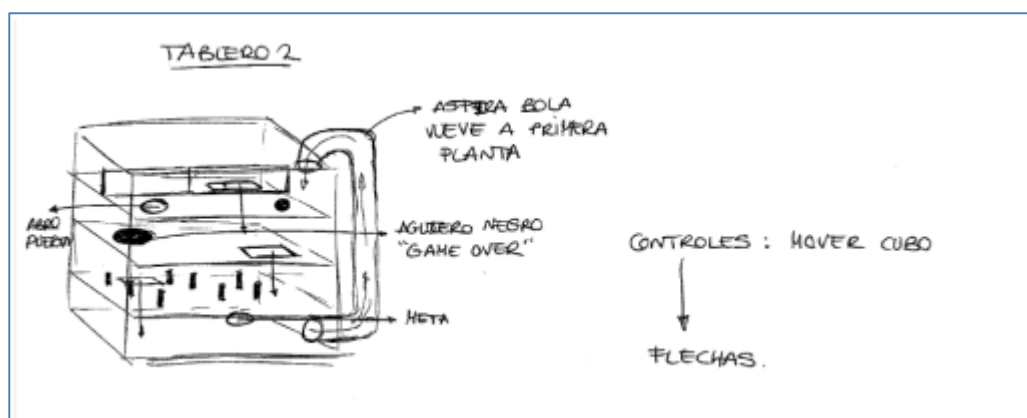


Ilustración 34. Story Board Tablero 2.

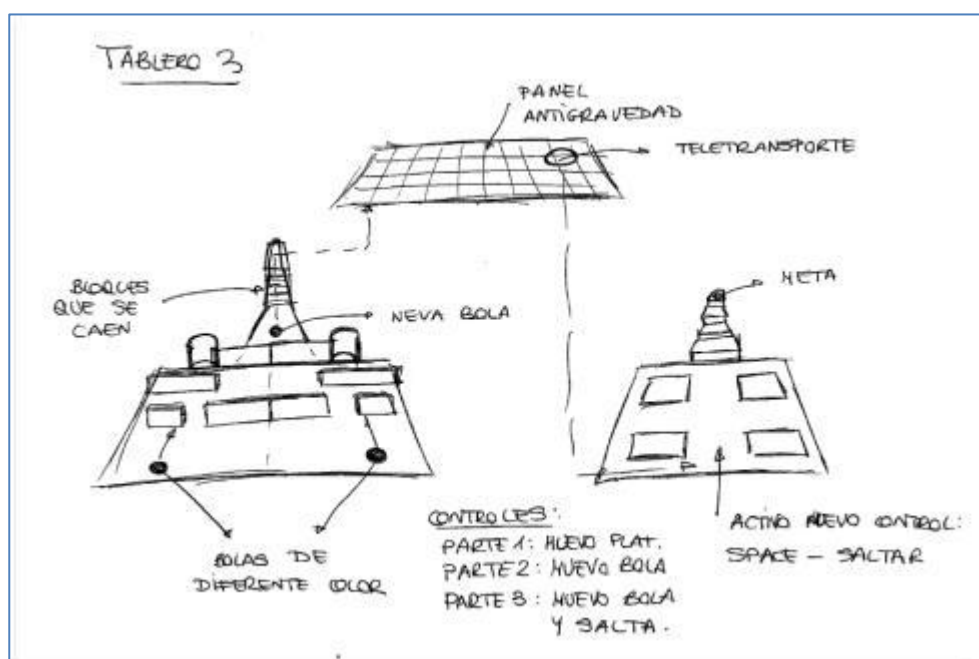


Ilustración 35. Story Board Tablero 3.

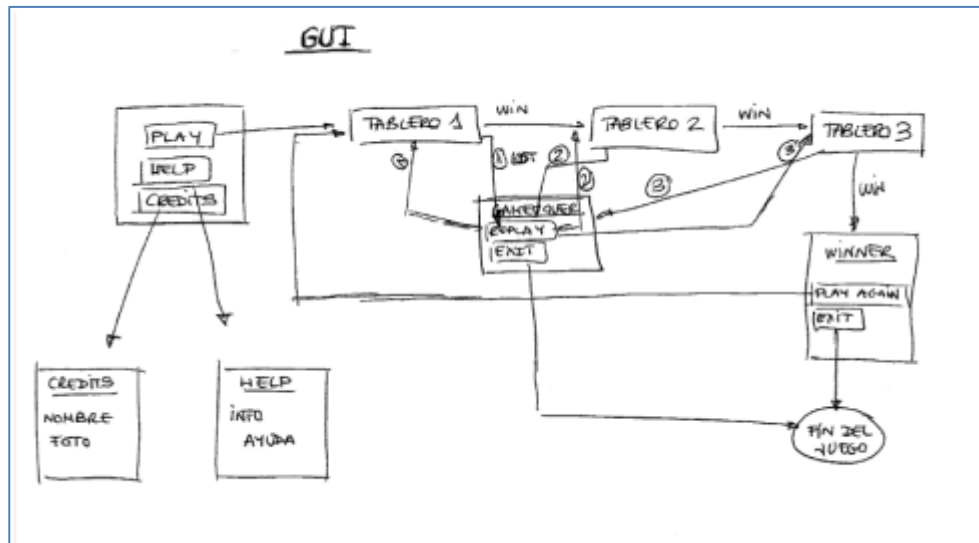


Ilustración 36. Story Board GUI.

En cuanto a la interfaz gráfica, también se emplea la misma técnica que para el diseño, se determinan el número de ventanas que se van a mostrar, las opciones dentro de cada una de ellas y la línea de ejecución como consecuencia de la interacción del usuario.

4.3.3. Sonido.

La elección de los sonidos es primordial para darle un aspecto más profesional, además de que, como veremos en el apartado 6.3, ayuda al aumentar la motivación y el interés de los jugadores durante la ejecución del juego. Muchas de las situaciones que se desencadenan en un videojuego a raíz de la interacción con el jugador, se acentúan y enfatizan mediante sonidos. Un sonido puede expresar alegría, emoción o miedo, y reproducirlo en el momento preciso hace que aumente la atención del jugador sobre este. En este videojuego se incluyen siete sonidos diferentes, sacados de una biblioteca de sonidos disponible en la red:

- **Reloj.** Se trata del sonido de un segundero. Para no crear frustración en el jugador se activa cuando faltan diez segundos para que la cuenta atrás del cronómetro finalice.
- **Bocina de un submarino.** Se trata del sonido de una bocina que se activa cuando la cuenta atrás finaliza.

- **Caída.** Se trata de un pitido que va disminuyendo de intensidad, por lo que el jugador lo asociará a un objeto que se aleja o, en este caso concreto, se cae. Se activa en cualquier situación en el que la esfera pierda el contacto con el plano que actúa como suelo en esa situación precisa
- **Teletransporte.** Se trata de un pitido intermitente. Se activa cuando la esfera sufre un teletransporte, es decir cambia de una posición a otra del escenario a través de uno de los mecanismos que se explican en el apartado 5.3.4.
- **Apertura.** Se trata de un sonido fuerte y constante que simula la apretura de una puerta electrónica. Se activa cuando el jugador usa uno de los mecanismos que se le proporcionan para abrir puertas que le impiden avanzar a lo largo del escenario.
- **Grito.** Se trata de un grito humano que se activa cuando la esfera colisiona con una de las trampas. El hecho de utilizar un sonido humano aumenta la interacción del jugador con su personaje virtual, ya que este no es más que un objeto cotidiano.
- **Victoria.** Se trata de un sonido alegre. Se activa cuando la esfera llega al mecanismo que considera que ese nivel ha sido superado.

4.3.4. Mecánica del juego.

Como se ha determinado en apartados anteriores, el juego consta de tres niveles, donde el personaje es una esfera que el jugador deberá guiar a través de escenarios tridimensionales hasta el final de cada uno de ellos. Antes de explicar los diferentes niveles, se van a detallar los controles que el jugador deberá usar, los mecanismos de que dispone para superar los niveles y los obstáculos que se encontrará:

Los controles, entendidos como las herramientas que se le facilitan al usuario para usar una aplicación, en este caso un videojuego, son los siguientes:

- Las flechas del teclado QWERTY. El movimiento desencadenado al pulsar estas teclas variará en función de objeto que tengamos que mover. Es cierto que el personaje es una esfera, sin embargo, el jugador no siempre tendrá que moverlo para llegar a otro espacio del escenario.

En el primer nivel, el usuario deberá guiar a la esfera, mediante los controles citados, por lo que el movimiento asociado a dichos controles se basará en un desplazamiento del personaje por el escenario.

En el segundo nivel, el jugador utilizará los controles para provocar una rotación en un cubo en el que se ven diferentes planos que dividen el cubo en varios niveles. Dentro del cubo se encuentra la esfera, por lo que la rotación de éste se traduce en un desplazamiento de la esfera por dichos planos.

Y por último, en el tercer nivel, se combinan los dos movimientos anteriores.

Por tanto, tomando como referencia los ejes de coordenadas X, Y y Z, y considerando que la posición desde la que observa el jugador viene dada por las coordenadas del punto naranja que se puede apreciar en la ilustración 30.

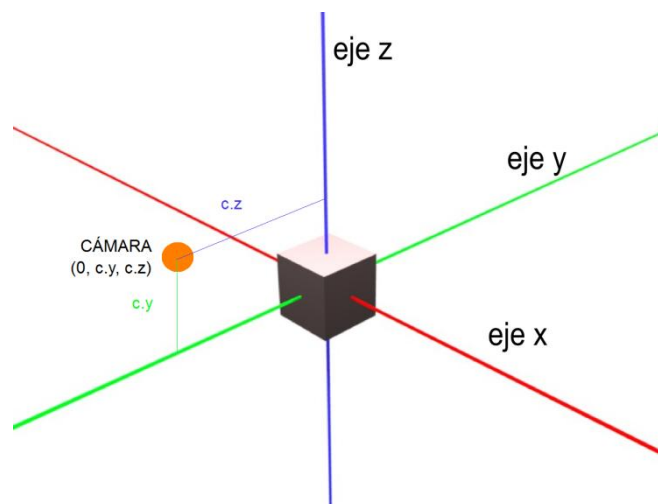


Ilustración 37. Coordenadas de referencia.

- “←”, desplazamiento de la esfera y rotación del cubo hacia valores negativos del eje X.
- “→”, desplazamiento de la esfera y rotación del cubo hacia valores positivos del eje X..
- “↑”, desplazamiento de la esfera y rotación del cubo hacia valores positivos del eje Y.
- “↓”, desplazamiento de la esfera y rotación del cubo hacia valores negativos del eje Y.

En el caso de que a lo largo del juego, se produzca un cambio de vista. Los controles se mantienen de la misma forma.

- La tecla “SPACE” del mismo teclado, para que la esfera sufra un desplazamiento en valores positivos del eje Z. Este control se activará en el tercer nivel.

Los mecanismos que dispone el jugador para superar los obstáculos que encontrará en los niveles, se traducen visualmente en planos circulares de diferentes colores que se encontrará a lo largo del juego. Como se explica a continuación, en función del color que tengan, se desencadenará una situación u otra:

- **ROJO.** Cuando el jugador guie a la esfera hacia un círculo de este color, se desencadenará un teletransporte, es decir, la esfera será llevada a una posición distinta dentro del escenario para poder continuar con el desarrollo del nivel.
- **VERDE.** El jugador deberá guiar a la esfera hacia un círculo de este color para poder abrir puertas que le impiden el paso dentro del nivel.
- **AZUL.** El jugador deberá guiar a la esfera hacia este círculo para poder cambiar la dimensión de ésta, ya que en caso de no hacerlo no podrá superar el nivel.
- **MORADO.** El círculo morado le indica al jugador que el nivel ha sido superado y por tanto, deberá guiar a la esfera hacia él para pasar al siguiente.

Además, del reto que se le impone al jugador de mantener la esfera dentro de las plataformas sin que sufra una caída, Se han diseñado diversos obstáculos para aumentar la intensidad y dificultad del juego:

- **Bloques fantasma.** Hay plataformas que visualmente parecen objetos sólidos, pero que tienen la propiedad “ghost”, lo que hace que la física en estos objetos no se tenga en cuenta, lo que supone que, en el caso de que el jugador guie a la esfera hacia esos bloques, ésta sufrirá una caída y en consecuencia, una penalización para el jugador, comenzar de nuevo dicho nivel.

- **Círculo negro.** Si el jugador guía a la esfera hacia un plano circular de este color, supondrá la inmediata penalización de comenzar de nuevo dicho nivel.
- **Antigravedad.** En un momento determinado del juego, la esfera sufrirá un cambio en la gravedad al ser dirigida hacia una de las plataformas del tercer nivel. Esto supone una inversión de las leyes físicas que rigen en el escenario, provocando un cambio circunstancial en la percepción del jugador.
- **Cinta transportadora.** En el momento en el que el jugador guíe a la esfera hacia esta cinta, será arrastrada sin ninguna escapatoria hacia el vacío, lo que supone la penalización inmediata de comenzar de nuevo el nivel para el jugador.

Una vez explicado esto, se van a describir más exhaustivamente los diferentes niveles:

- **Primer nivel.** El jugador parte de una plataforma en forma de “S”, por la que debe guiar al personaje hasta llegar a otra plataforma que está compuesta por diferentes cubos de colores, donde algunos son bloques fantasma. Este debe encontrar el camino correcto con el fin de llegar hasta una rampa que le conduzca a una plataforma que se encuentra más abajo.

Una vez en la plataforma inferior, deberá guiar a la esfera hacia el plano circular de color verde, que como se ha explicado anteriormente, permite abrir la puerta que impide la continuación del juego, ya que la meta se encuentra al otro lado de ésta.

- **Segundo nivel.** El jugador se encuentra dentro de un cubo transparente dividido por varios planos que simulan diferentes pisos dentro de éste. Parte desde el del piso superior y debe llegar al inferior guiando a la esfera hacia los orificios que hay en cada plano. En este nivel, el jugador no controla el movimiento de desplazamiento de la esfera, sino el movimiento de rotación del cubo, donde la esfera, desde un punto de vista físico, es un cuerpo muerto que se mueve debido a la inclinación de los planos contenidos en el cubo.

En el primera plano, debe rotar el cubo con el fin de conducir la esfera hacia el plano circular verde para abrir la puerta que le impide el paso hacia

la apertura que se encuentra al otro lado, y que le conduce hacia el plano inferior.

En el segundo plano, debe evitar que la esfera toque el plano circular negro y llegar a al orificio que se encuentra al fondo del plano.

En el tercer plano, el jugador debe observar bien los elementos que se le muestran alrededor. Por un lado, tiene que sortear la cinta transportadora para evitar la penalización, por otro, percatarse del plano circular azul que le ayudará al cambio de tamaño de la esfera, ya que si no, no conseguirá superar el nivel, y todo ellos para guiar a la esfera hacia el agujero que le conduce al nivel inferior.

Y en el cuarto y último, en el caso de que haya conseguido cambiar las dimensiones de la esfera, podrá introducirla en el pasillo formado por paredes translucidas, para así conducirla a la meta; y en caso contrario deberá dirigir la esfera hacia el plano circular rojo, que le hará teletransportarse hasta la primera planta, penalizándole a comenzar de nuevo.

- **Tercer nivel.** En este nivel, se combinan los dos movimientos que el jugador efectúa a partir de los controles facilitados, el desplazamiento de la esfera y la rotación de un objeto para provocar el movimiento de ésta.

El tercer nivel se inicia en una plataforma donde el jugador debe conseguir llevar dos esferas, de distinto color, mediante la rotación de la plataforma que las contiene hacia las metas correspondientes, diferenciadas por el color de las esferas. Una vez terminado este objetivo, habrá un cambio de escena, un cambio en los controles y un cambio de personaje.

En esta segunda fase del tercer nivel, el personaje es otra esfera, de color diferente a las dos primeras, la cual debe guiar por la plataforma en la que se encuentra y donde sufrirá un nuevo cambio de escena. Este cambio introduce un nuevo elemento del juego, lo que en el punto anterior se ha denominado *antigravedad*. La esfera es elevada al final de la plataforma, sin que el usuario pueda interactuar con ella para evitarlo, y colocada en otra plataforma que tiene esta característica, y en la cual debe encontrar la forma de invertir de nuevo la dinámica del juego. El mecanismo facilitado para ello es el *teletransporte*, es decir, el jugador debe dirigir la esfera hacia un plano circular de color rojo, que en esta ocasión no está sobre la

plataforma que simula el suelo, sino rotando dentro de ésta. Una vez consiga dirigir al personaje hasta allí, se producirá otro cambio de escena que le llevará a la última fase de este nivel.

En esta última fase, el jugador se encontrará con una nueva plataforma que tiene una característica adicional, ya no se trata de una plataforma con la superficie plana, sino que ésta se encuentra modelado mediante planos inclinados combinados con orificios. Debe ser capaz de sortear estos obstáculos para llegar a la escalera que se encuentra al final de la plataforma, donde se acciona un nuevo control, la tecla *SPACE* del teclado, para que la esfera reciba la acción de saltar, y el jugador llegue a la meta que se encuentra al final de la escalera.

4.3.5. Diseño de las plataformas.

La herramienta de modelado que ofrece Blender es bastante sencilla e intuitiva, proporciona diversos objetos, que mediante distintas transformaciones, permiten diseñar escenarios más que aceptables. Hay que destacar, que Blender es en sí una aplicación de modelado 3D, al que se le pueden integrar diversos motores de juego, para el desarrollo de videojuegos. En este caso en concreto, el motor utilizado es el que proporciona la aplicación, el motor *Blender Engine*, pero eso no nos impide la integración de otros de mayor calidad.

El hecho de haber decidido utilizar el motor propio de Blender aporta ventajas y acarrea algunas limitaciones. Las ventajas a destacar son, en primer lugar no tener que aprender el manejo de una herramienta; en segundo lugar, no tener que realizar un trabajo de integración entre el motor de juego y Blender, evitando aspectos como la instalación, la configuración o la incompatibilidad con el sistema operativo o la versión de este; y en último lugar, utilizar una herramienta *open source*. Sin embargo, aparecen varias limitaciones, la primera es que muchas de las *features* que nos proporciona la herramienta de modelado Blender no son compatibles con su ejecución en el motor de juego, cosa que nos limita sobre todo a la hora de crear objetos de formas suavizadas; y la segunda es que renunciamos a la ejecución en un motor de juego más potente.

La fase de diseño de las plataformas, se inicia a continuación de la etapa de arte conceptual, donde se realiza un trabajo artístico y creativo que marcará la idea de esta fase.

Blender dispone de una ventana de modelado 3D, que aparece por defecto al ejecutar el programa, en ella aparece un plano guía dividido en una cuadrícula para marcar un plano de referencia al comenzar a trabajar. En él se marcan los ejes de coordenadas bidimensionales, el eje X en rojo y el eje Y en verde y un cursor 3D, representado por un círculo, que sirve como referencia a la hora de crear los objetos. El eje Z, representado visualmente mediante una línea de color azul, será visible en el momento en el que se empiecen a crear los objetos

También es importante saber que Blender dispone de varias vistas que se pueden elegir en la barra de herramientas o mediante atajos de teclado. Entre ellas, las vistas básicas a la hora de realizar cualquier dibujo en perspectiva, alzado, planta y perfil, y otras como campo visual visible desde las diferentes cámaras que se encuentren en el escenario. Además, se puede cambiar la vista mediante el ratón, pudiendo colocar la vista en cualquier espacio tridimensional alrededor del modelado.

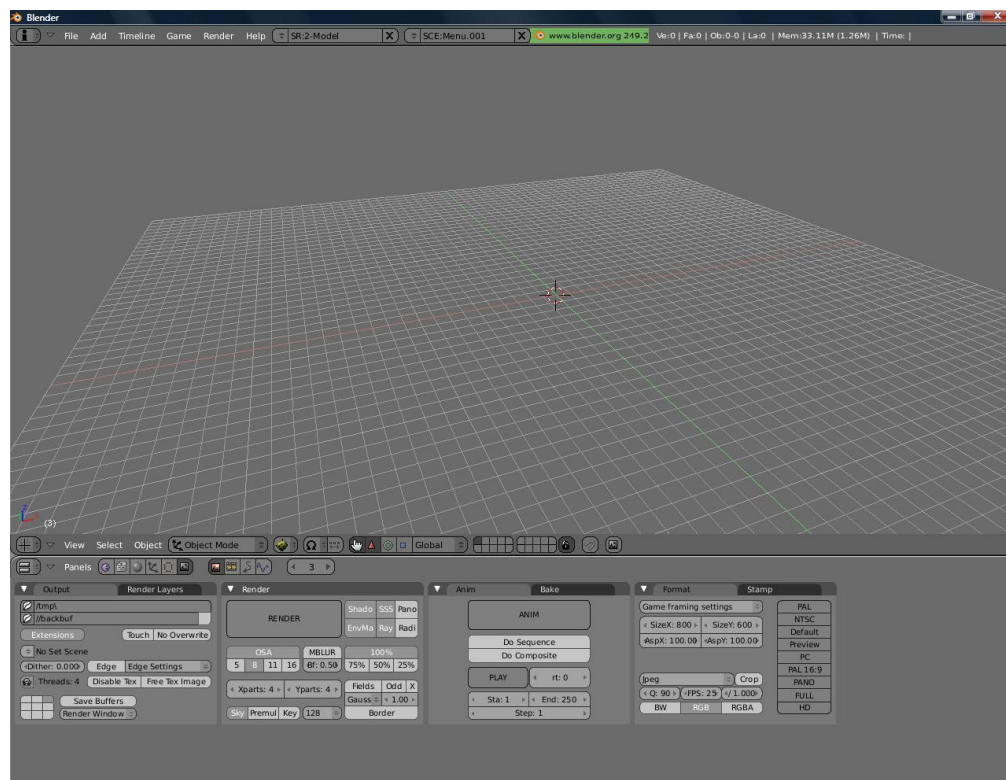


Ilustración 38. Ventana por defecto.

Hay multitud de objetos que se pueden utilizar para comenzar a realizar el modelado. La barra de herramientas, dispone de una opción *Add* para elegir el objeto que más se adapte al diseño. Debido a la limitación surgida a raíz de la elección del motor de juego durante el modelado, los objetos que utilizados en

este caso, son los llamados objetos *Mesh*, objetos malla, dentro de éstos, disponemos de:

- **Planos**, objeto de cuatro vértices, cuatro aristas y una cara. Por sí solo no se trata de un objeto tridimensional. En este caso, se ha utilizado para modelar los suelos y las paredes de algunos de los niveles.
- **Cubo**, objeto tridimensional por sí solo, utilizado en los escenarios para la realización de las plataformas, tanto para las que tienen superficie plana como para superficies inclinadas, de los bloques fantasmas y los objetos que simulan puertas.
- **UVSphere**, esfera compuesta por n segmentos y m anillos, donde el valor de estas variables indicará el nivel de detalle que se quiere, valor que se elige en una ventana emergente que aparece cuando se crea este objeto. Aumentar estos valores también influye en la carga computacional de renderizado de la escena 3D. Los segmentos son como los meridianos terrestres que van de polo a polo y los anillos como los paralelos. Este objeto se ha utilizado para el personaje principal del videojuego.
- **Cilindro**, hecho a partir de una sección circular de n vértices que se determinan en la ventana emergente que aparece en su creación. Las metas de la primera plataforma del tercer nivel están modeladas a partir de este objeto.

Además de estos objetos básicos, se dispone de otros muy importantes en el desarrollo, no sólo de videojuegos sino también en animación:

- **Cámaras**, en este caso, hay niveles en los que se han utilizado hasta cuatro cámaras. El uso más frecuente en el diseño de videojuegos, es el cambio de la perspectiva de la visión del jugador.

Cada vez que se produce un cambio de cámara, inevitablemente, se desencadena un cambio en los controles del usuario. En el caso de la primera y de la tercera pantalla, únicamente se realizan dos cambios de escena, por lo que sólo se han implementado dos cambios en los controles. Sin embargo, en el segundo nivel, se producen cuatro cambios en la perspectiva, con lo que es necesaria la implementación de cuatro cambios en los controles.

- **Lámparas**, la iluminación puede hacer que varíen muchos aspectos de la animación. Blender dispone de cuatro tipos de luces: *solar*, *hemi*, *lamp* y *spot*. La diferenciación entre ellas se encuentra en aspectos como la intensidad, la dirección de proyección de los rayos y la sombra que proyectan. Para este desarrollo se han empleado los siguientes tipos de lámparas:
 - **Luz solar**, para la iluminación general de todos los niveles, se trata de una lámpara colocada en el centro de gravedad del modelo a una altura considerable para que las imágenes especulares reflejadas y las sombras sean de una intensidad menor.
 - **Luz lamp**, para la iluminación en el segundo nivel, ya que al haber diferentes plantas, es necesario una iluminación en cada una de ellas. Tiene una intensidad considerablemente menor que la solar.
 - **Luz spot**, lo que en castellano se denomina luz focal, para la iluminación del personaje en cada nivel. Se ha colocado este tipo de lámpara encima de cada esfera y se ha implementado que efectúe los mismos movimientos que ella, con el fin de que esté permanentemente iluminada.

Partiendo de los objetos que hemos explicado anteriormente, Blender proporciona varias transformaciones que se pueden aplicar a todos ellos, y con las que se puede obtener objetos más moldeados y refinados.

Las tres más importantes son: la rotación, el desplazamiento en el plano de referencia y el dimensionado; se pueden realizar en los tres ejes de coordenadas, de manera conjunta o de forma independiente. De esta manera podemos transformar un simple cubo en un hexaedro irregular donde una de sus caras y su contraria no tengan las mismas dimensiones y que además, ésta tenga una rotación en cualquiera de los tres ejes. Lo podemos apreciar mejor en la ilustración 32, en el diseño de la derecha se ha realizado a partir del conjunto de varias transformaciones disponibles en Blender sobre un objeto primario .

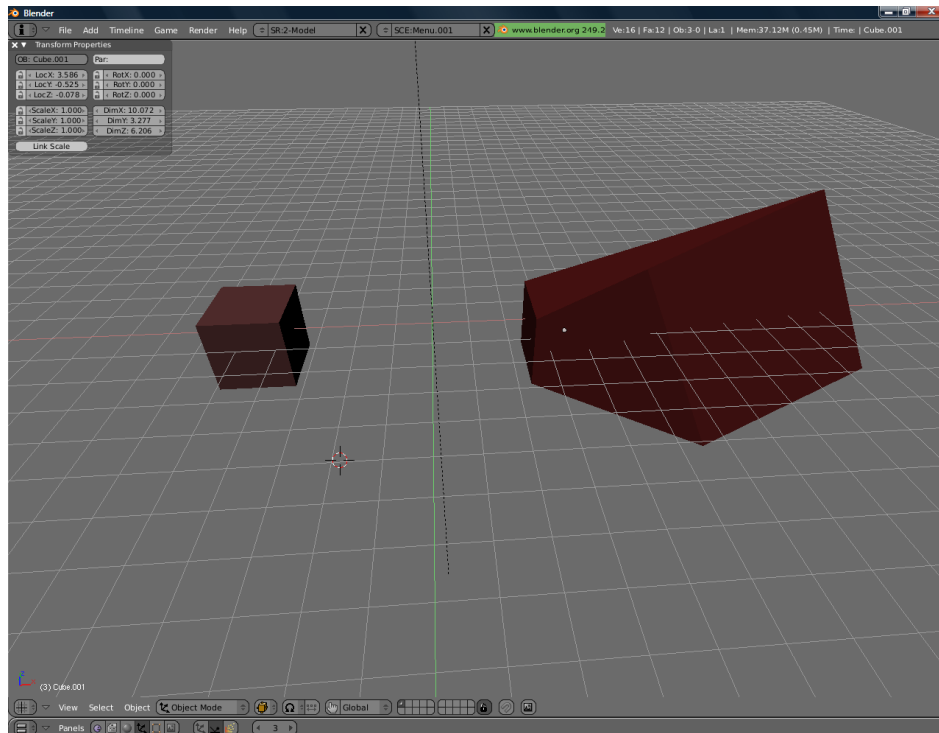


Ilustración 39. Transformaciones básicas.

Otras de las herramientas que Blender nos proporciona entrañan una complejidad mayor. Es posible duplicar los objetos; dividir planos; rotar, desplazar y redimensionar uno o varias caras, sus vértices o sus aristas; eliminar o añadir caras, vértices o aristas; entre otras.

Una vez que se ha realizado el modelado de cada uno de los niveles, se comienza con la decoración. Para ello, Blender dispone de dos ventanas, la ventana de materiales y la ventana de texturas. En la ventana de materiales se cambian características estéticas como el color del objeto dependiendo del tipo de luz que le incida, o la transparencia del objeto. En la ventana de texturas, se pueden aplicar filtros de ruido para generar diferentes texturas o aplicar la texturización a partir de una imagen.

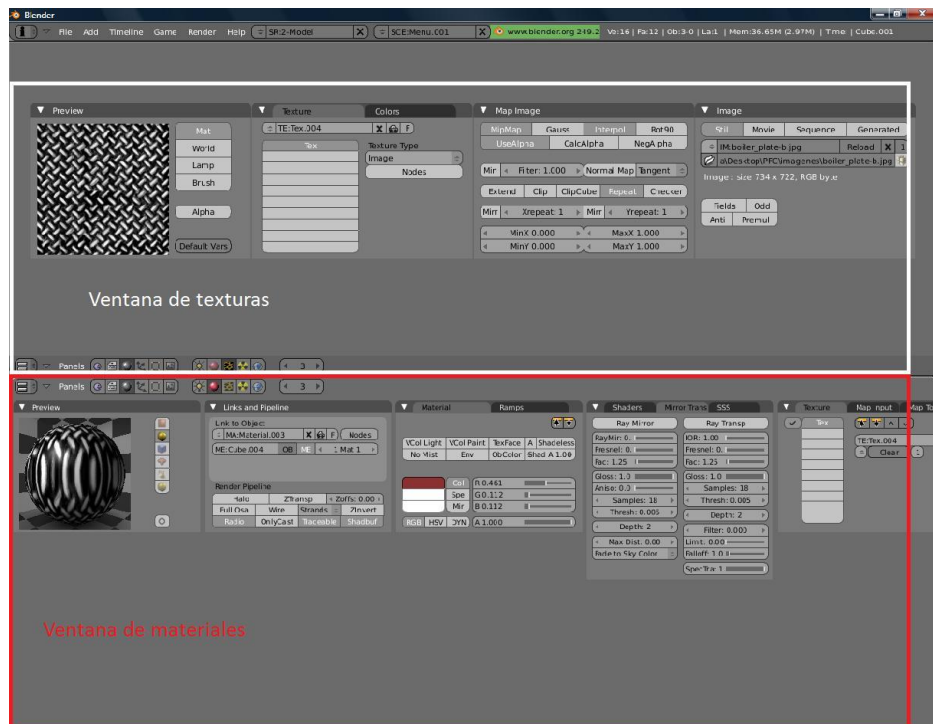


Ilustración 40. Ventanas de materiales y texturizado.

Después de esta breve explicación del desarrollo del modelado, vamos a ver gráficamente el trabajo realizado.

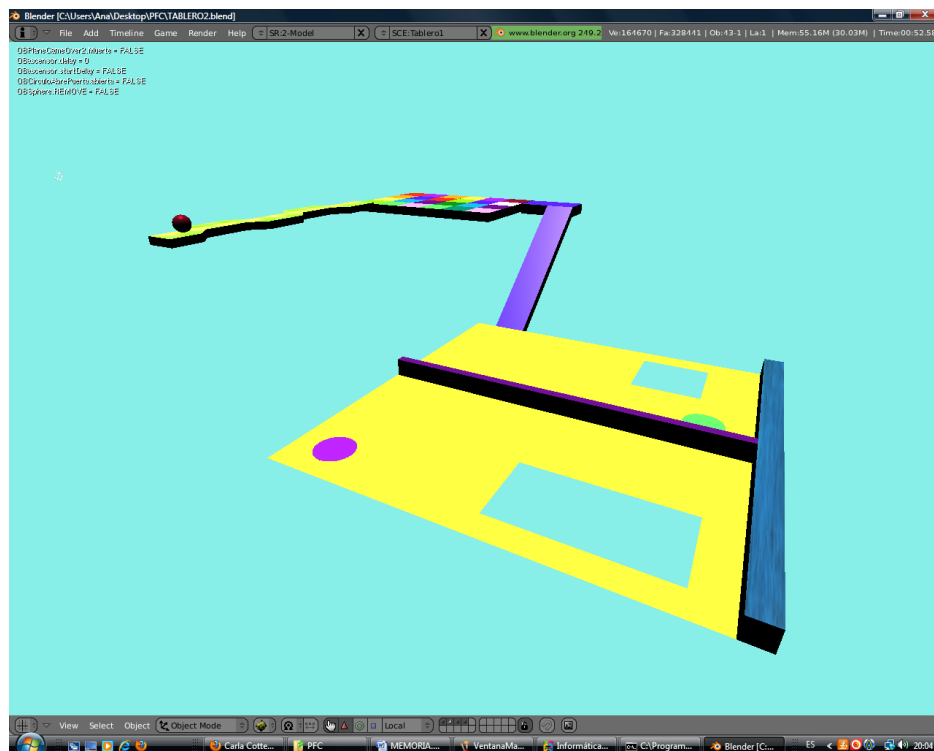


Ilustración 41. Nivel 1.

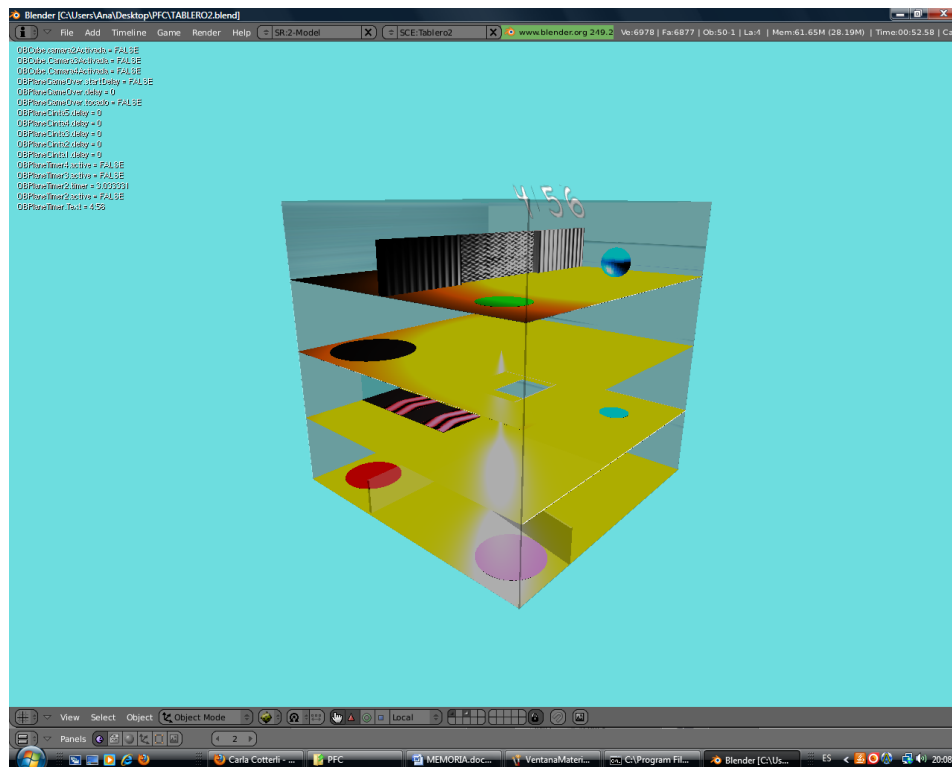


Ilustración 42. Nivel 2.

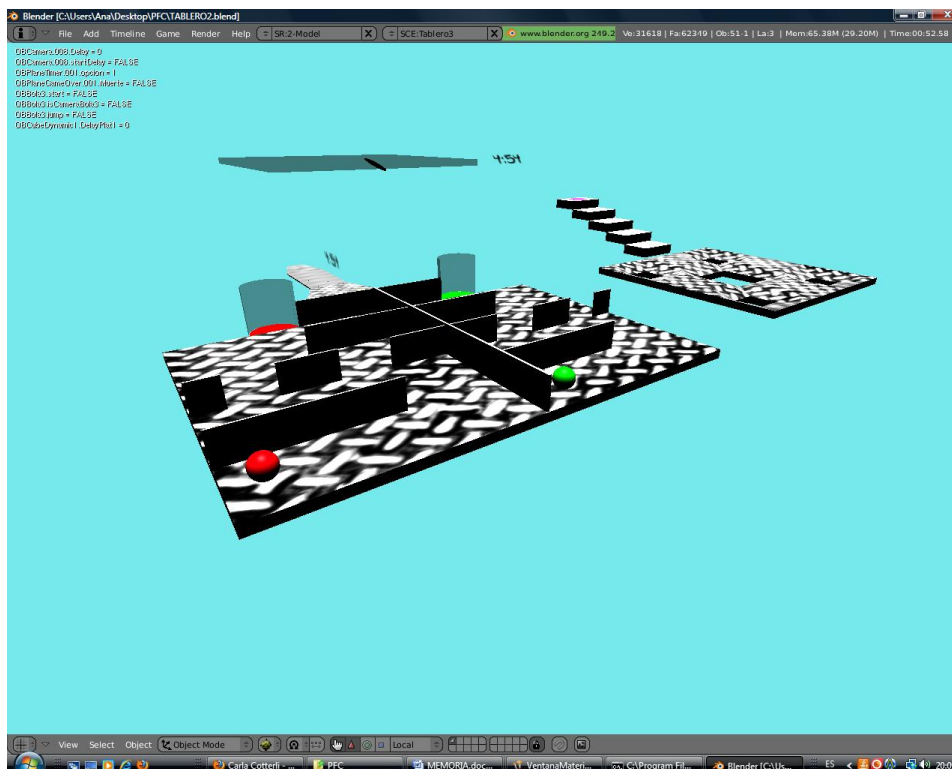


Ilustración 43. Nivel 3.

Como se puede apreciar, el modelado y texturización en Blender es muy costoso. Hay que tener en cuenta muchos parámetros y saber utilizar bien las herramientas que te proporciona para sacarle el mayor partido a la aplicación.

4.3.6. Diseño de programación.

Como se ha explicado en el apartado “Blender Engine” (3.3), una de las facilidades que nos proporciona Blender en el diseño de videojuegos, es que disponemos de una ventana para el diseño de la lógica. En ella, se programa de manera visual cada objeto, tanto si en el videojuego tiene un papel estático como dinámico. Pero también, proporciona la posibilidad de que el diseñador implemente funciones adicionales mediante *scripts* escritos en *Python*.

Para entender un poco mejor la forma visual de implementación, partimos de la explicación de tres conceptos, sensores, controladores y actuadores. Hay que tener claro el desarrollo lineal de la implementación, la comunicación entre los sensores, controladores y actuadores se realiza mediante pulsos que se envían al finalizar la acción especificada por el programador en cada uno de ellos. De tal manera que el ciclo de vida de una implementación determinada para un objeto concreto queda ilustrado en la imagen número 37.

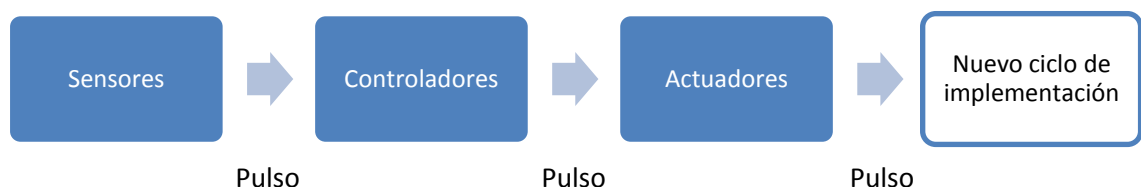


Ilustración 44. Diagrama de ciclo de vida.

Como se aprecia en la ilustración 37, los sensores envían un pulso a los controladores en el momento en el que ha detectado alguna acción que el programador haya configurado en ellos, estos a su vez, envían un pulso a los actuadores cuando determinan que la señal enviada por los sensores es la que se desea capturar, y por último, los actuadores realizan la acción que se ha configurado en ellos, y realimentará un nuevo ciclo de implementación si el programador así lo desea.

A continuación, se va explicar más detalladamente la función de cada uno de ellos y los tipos de configuración seleccionables.

- **Sensors.** Se puede definir “*sensors*” como los mecanismos que ofrece Blender para la captura de diversas situaciones que se puedan producir dentro de la ejecución de un videojuego. Hay varios tipos de sensores, para la implementación de la lógica de este juego se han utilizado los siguientes:
 - *Keyboard.* Se trata de un sensor que permite capturar los eventos de teclado. En el caso que nos concierne, se ha utilizado para capturar el evento de teclado de las flechas y de la tecla SPACE.
 - *Message.* Con este sensor se captura cualquier mensaje que un actuador haya enviado para iniciar una nueva implementación como consecuencia de otra. Este sensor es necesario por la forma que proporciona *Blender* para la implementación, puesto que esta se realiza de manera independiente para cada objeto.

Para que sea más fácil de entender, se va a ilustrar con un ejemplo. Como se ha explicado en el apartado 5.3.4., donde se define la mecánica del juego, el jugador debe colocarse encima de un círculo verde para poder abrir una puerta que le impide el paso. Claramente se ve la interacción entre dos objetos distintos, que tendrán dos implementaciones diferentes según establece la filosofía de *Blender*. Cuando la bola colisione con el círculo, implementación incluida dentro de un ciclo completo, el actuador del ciclo de implementación enviará un mensaje al sensor del objeto puerta para iniciar un nuevo ciclo que implementará a apertura del objeto puerta.

- *Property.* Los objetos pueden diferenciarse unos a otros definiendo propiedades, propiedades que se definen mediante este sensor. Con este sensor se pueden realizar comprobaciones como saber el valor de variables que puede ser booleanas o numéricas. Un caso de uso dentro de este proyecto es evaluar una comprobación para saber si la variable “*abierta*” de tipo booleana en un objeto puerta tiene el valor *true* o *false*, para en función de este valor efectuar una acción u otra.
- *Collision.* Mediante la utilización de este sensor se determina si un objeto, asociado a él, ha colisionado con otro caracterizado por una propiedad determinada. A continuación se ilustra con un ejemplo. Los objetos esféricos, que forman el personaje, tienen asociada una propiedad de nombre “bola”. En los objetos que son planos

circulares de color verdes hay un sensor de colisión en el que se define que, cuando se detecte una “collision” con la propiedad “bola”, envíe un pulso al controlador correspondiente.

- *Always*. Este sensor se utiliza para acciones se tienen que ejecutar durante todo el transcurso del juego. En este proyecto, se ha utilizado, por ejemplo, para la ejecución de un *script python*.
- **Controllers**. Los controladores son principalmente operaciones booleanas básicas para controlar uno o más sensores, aunque también se dispone de un controlador especial para la ejecución de los *scripts*.
 - Las operaciones booleanas disponibles son XNOR, NOR, NAND, XOR, OR y AND. Se utilizan multitud de veces para determinar la salida final de la agrupación de varios sensores.
 - *Script*. Con este tipo de controlador se ejecutan los script que se hayan implementado. En este caso, se han implementado varios *scripts*, por ejemplo, en uno de ellos se ha programado la cuenta atrás.
- **Actuators**. Mediante los actuadores, se añaden todas las funcionalidades que se quieran a los objetos que forman el videojuego, en función de las capturas que recogidas a partir de los sensores y que de alguna manera se han seleccionado gracias a los controladores.
 - *Parent*. Gracias a este actuador se puede eliminar o crear una relación de parentesco entre dos o más objetos, determinando cuál de ellos es el padre del grupo. De esta manera un movimiento aplicado a un objeto padre será reproducido de manera similar en los objetos hijo. En la implementación de este videojuego, se han utilizado varias relaciones de parentesco, un caso en concreto es la relación entre el objeto esférico y la cámara asociada a él, donde la esfera se convierte en el objeto padre y la cámara en el objeto hijo, de tal manera que la cámara copiará cualquier movimiento que haga la esfera.
 - *Game*. Mediante este actuador se controlan aspectos de la ejecución del juego, como es su finalización, su reiniciación, iniciar otro juego distinto, o cargar librerías de ejecución nuevas. Un ejemplo claro en el desarrollo del videojuego es cuando el jugador

selecciona la opción “exit” en el menú de la interfaz gráfica, lo que supone la inmediata finalización de la ejecución del videojuego.

- *Message*. De esta forma se envía un mensaje para que uno o más sensores lo puedan capturar. Como se ha explicado en el punto donde se definen los diferentes sensores, el plano circular de color verde le envía el mensaje “openDoor” y un sensor asociado al objeto puerta captura dicho mensaje.
- *Property*. Con este actuador, se puede cambiar el valor de las propiedades del objeto asociado a este actuador. En el caso de querer cambiar las propiedades de otro objeto desde éste, se debe mandar un mensaje a dicho objeto para que el sensor de éste lo pueda capturar y mediante un actuador de este tipo asociado a él cambiar el valor de la propiedad.
- *Sound*. Este actuador se utiliza para reproducir audio en el videojuego. La manera de reproducirlo es, mediante archivos de música de tipo *mp3* almacenados en disco, especificar la ruta del archivo *mp3* que queremos reproducir.
- *Scene*. Este actuador se utiliza para activar o desactivar la visión de una cámara o realizar un cambio de escena. En este proyecto, se ha dividido la ejecución del juego en escenas, es decir, cada nivel se ha diseñado en una escena distinta, así como las pantallas del menú de la interfaz gráfica. Por tanto, la transiciones entre ellas se efectúan mediante este actuador.
- *Motion*. Se encarga de aplicar movimiento al objeto asociado a él a partir de unos parámetros, a los que se le asignan determinados valores en función del movimiento que se quiere implementar. Estos parámetros controlan el desplazamiento, la rotación, la fuerza, el esfuerzo de torsión, la velocidad lineal y la velocidad angular, en cada eje (X, Y, Z).
- *IPO*. Mediante este actuador se reproduce el movimiento de un objeto mediante una curva IPO asociado a él. Se utiliza para añadir movimiento a un objeto sin que haya un control del jugador sobre éste, sino que forma parte del entorno dinámico del videojuego. Los movimientos que se pueden realizar son el desplazamiento, la rotación y el escalamiento del objeto en los tres ejes.

En este caso, las curvas IPO han sido muy utilizadas, por ejemplo y siguiendo con lo explicado anteriormente, el movimiento que realizan los objetos puerta a raíz de la colisión del objeto esférico con el plano circular de color verde, se realiza mediante una curva IPO muy sencilla, ya que, únicamente, se produce un desplazamiento en el eje X. Sin embargo, la implementación del movimiento del personaje en los teletransportes también está realizada con curvas IPO en las que se capturan desplazamientos en los tres ejes.

4.3.7. Ejemplos de la implementación

Para obtener una mejor visión de la implementación, se va a proceder a continuación a realizar una explicación gráfica de los tres mecanismos de implementación a partir de los ejemplos descritos anteriormente:

- **Ejemplo 1.**

En la ilustración 38 se muestran las dos ventanas que se necesitan para la implementación en *Blender*, la ventana de modelado y la ventana de implementación, marcada en rojo. La zona remarcada en rojo, muestra el objeto que se está implementando.

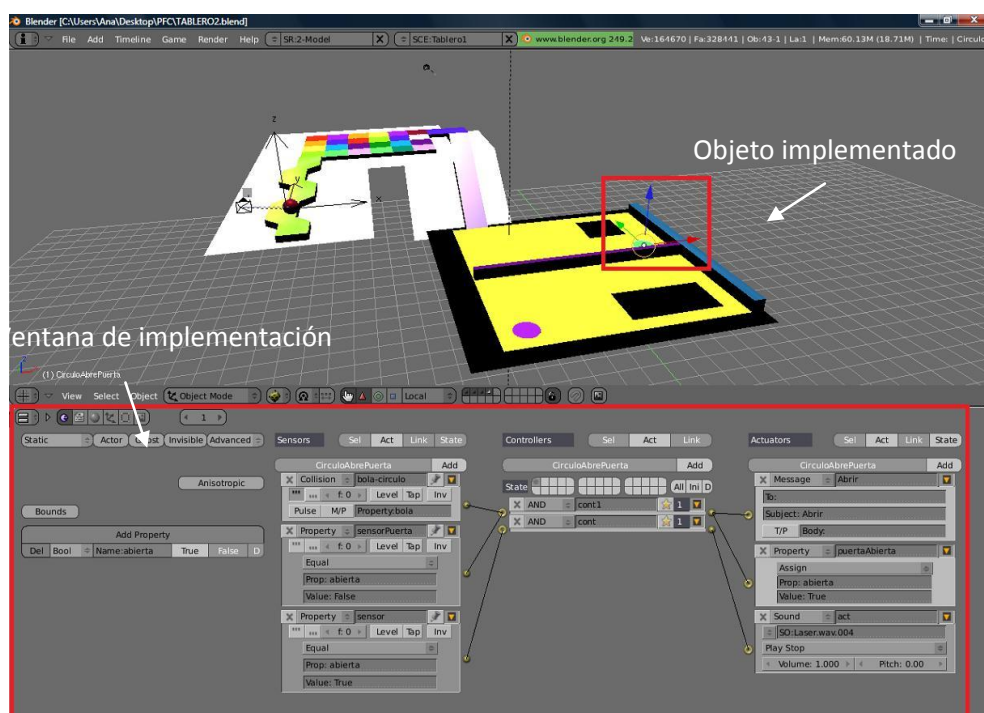


Ilustración 45. Blender workspace ejemplo 1.

En la ilustración 39, se ve ampliada la ventana donde se implementan las acciones de cada objeto.

- Los recuadros azules resaltan las herramientas explicadas anteriormente, *sensors*, *controllers* y *actuators*.
- Los recuadros rojos resaltan los diferentes tipos que se han utilizado para esta implementación en concreto, los sensores *collision* y *property*, el controlador *AND* y los actuadores *message*, *property* y *sound*.
- El recuadro amarillo indica el nombre del objeto que se está implementando, *circuloAbrePuerta*.
- El recuadro verde indica la propiedad asociada a ese objeto, se trata de una variable booleana de nombre *abierta* y de valor inicial *False*.

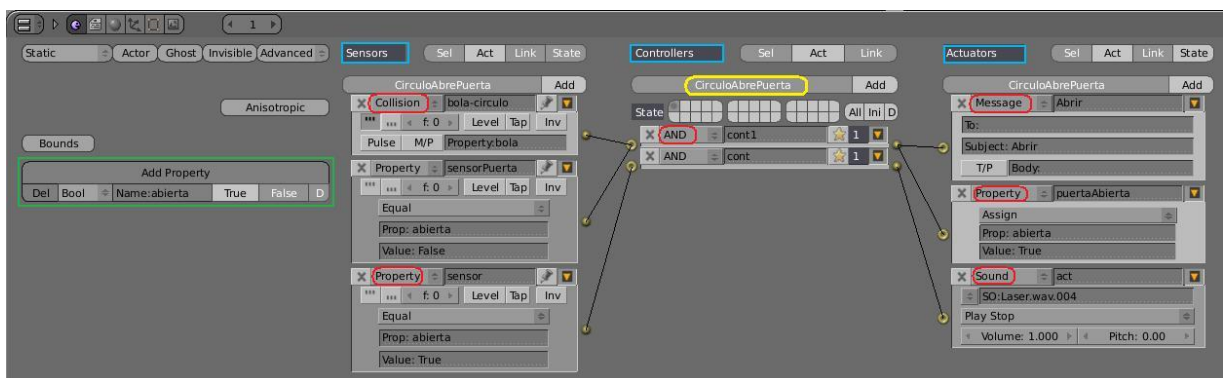


Ilustración 46. Ventana de implementación del objeto *CirculoAbrePuerta*.

Ampliando la zona en la que se encuentra la implementación en la ilustración 39, se ve claramente que hay dos ciclos de implementación.

En el primero están involucrados dos sensores, un controlador y dos actuadores. Cuando el objeto esférico que viene definido por una propiedad llamada *bola* colisione con el objeto *circuloAbrePuerta* y, siempre que, la propiedad *abierta* tenga el valor *False*, se enviará un mensaje con el nombre *Abrir*, que iniciará el segundo ciclo de implementación, y a su vez, cambiará el valor de la propiedad *abierta* al valor *true*. Hay que destacar que si un sensor de cualquier otro objeto escuchara el mensaje enviado, se iniciaría el ciclo de implementación asociado a dicho objeto.

En este segundo ciclo están involucrados un sensor, un controlador y un actuador. Cuando el sensor *property* detecte que la propiedad *abierta* ha cambiado de valor a *true* enviará un pulso que recogerá el controlador y que, a su vez, enviará otro pulso que recogerá el actuador *sound*, iniciando así la reproducción del sonido, ya precargado, de nombre *Laser*.

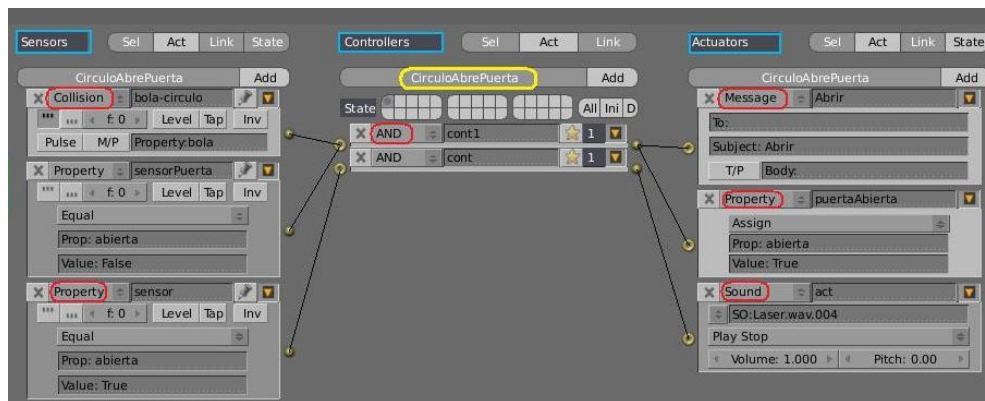


Ilustración 47. Sensors, Controllers & Actuators.

• Ejemplo 2.

A continuación, vamos a ver la implementación que se ha realizado para asociarle los controles para el movimiento del objeto esférico. En la ilustración 41, se observa el entorno de trabajo de *Blender*, con las correspondientes ventanas necesarias, la de modelado y la de implementación.

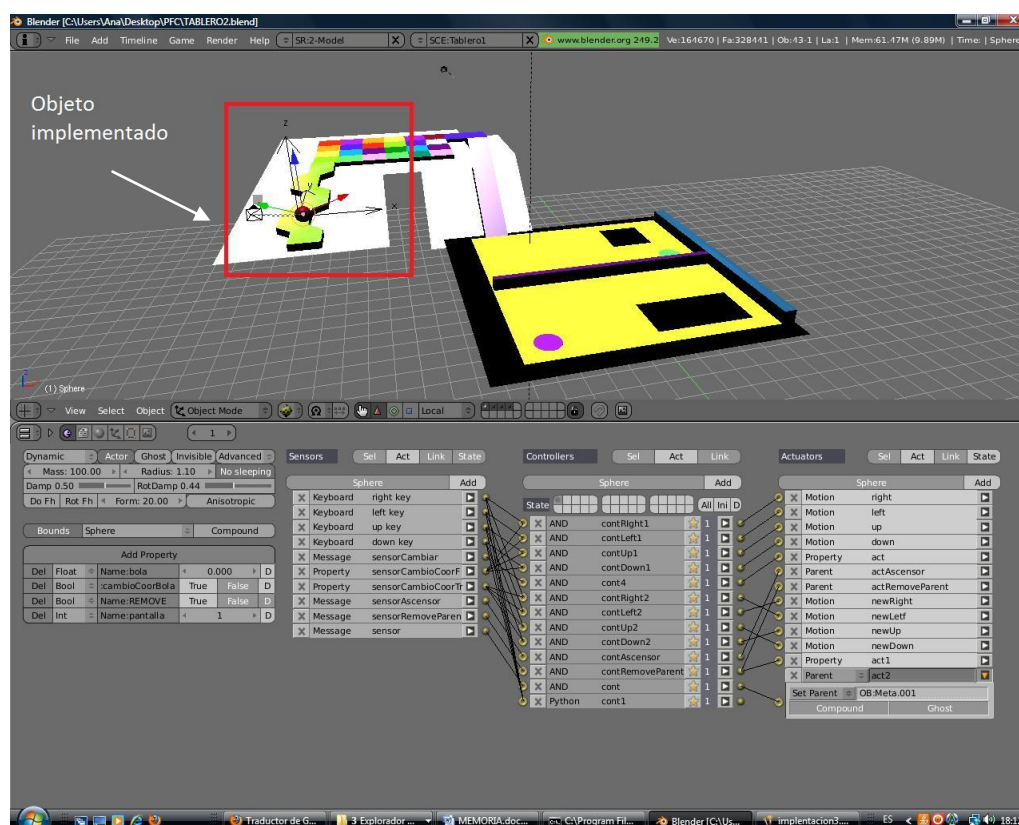


Ilustración 48. Blender workspaces ejemplo 2.

Siguiendo la dinámica utilizada en el ejemplo anterior, vemos que, resaltado mediante recuadros de color rojo en la ilustración 42, se han asociado, al objeto esférico, cuatro sensores *keyboard* y cuatro actuadores *motion*. Se

puede ver en dicha ilustración que se han implementado más ciclos aparte del descrito, esto se debe a los cambios de cámaras que se producen durante el juego, puesto que un cambio de cámara implica un cambio de sentido y dirección en la trayectoria del movimiento del objeto. Sin embargo, como su implementación es prácticamente igual, únicamente se va a explicar más detalladamente la implementación de un ciclo.

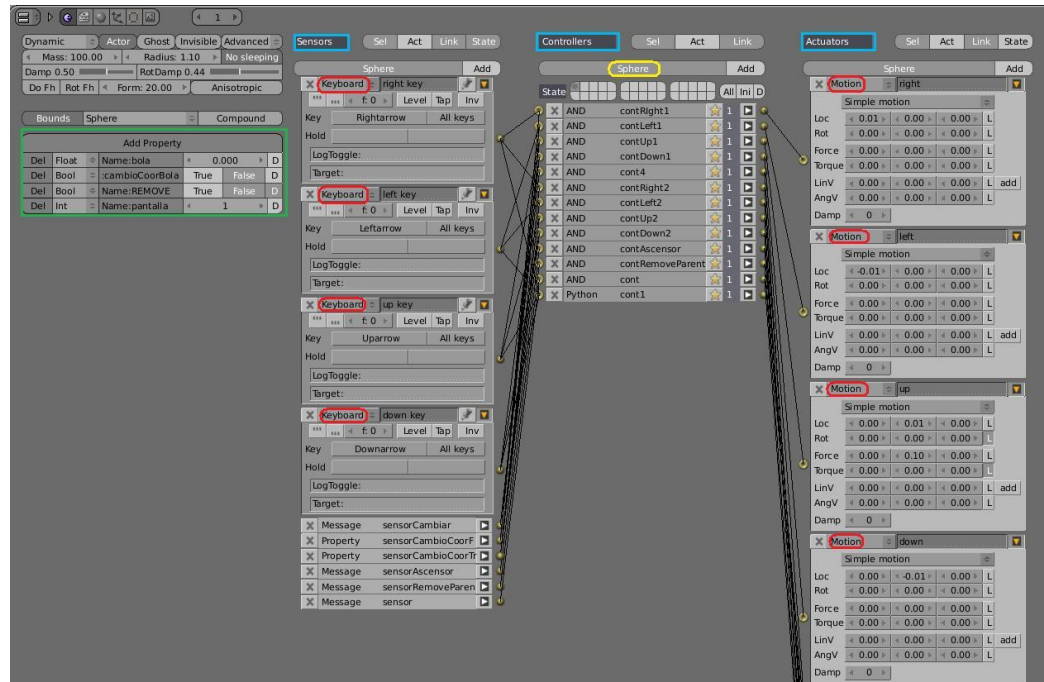


Ilustración 49. Ventana de implementación del objeto *Sphere*.

Por tanto, en la ilustración 43 se ve de manera ampliada lo que se quiere resaltar de esta implementación. En este caso, a pesar de dirigirnos a él cómo un único ciclo, en realidad se trata de cuatro ciclos de implementación distintos, diferenciados por el evento de teclado recogido por los sensores y por los diferentes movimientos seleccionados en los actuadores.

Por una parte, se ve claramente un ciclo básico compuesto por la sucesión lineal de la comunicación entre, el sensor *keyboard*, que recoge el evento de teclado, el controlador *AND* y el actuador *motion*. Este ciclo básico se descompone en cuatro ciclos más característicos que son los siguientes:

- *Ciclo 1. Movimiento hacia la derecha.* El sensor *keyboard* recoge el evento de teclado de la tecla “flecha hacia la derecha”, cuando lo detecta envía un pulso al controlador *AND*, y este a su vez, envía un pulso al actuador *motion* que aplicará una variación de 0.01, por cada pulso recibido, en la posición del objeto esfera en el eje X, representado en la venta de implementación como una variación en el valor de la primera columna de la variable *Loc*.

- *Ciclo 2. Movimiento hacia la izquierda.* El flujo del ciclo es exactamente el mismo que el del ciclo 1, las dos variación son, por un lado, el evento de teclado recogido es el de la tecla “flecha hacia la izquierda”, y por otro, la variación de -0.01, por cada pulso recibido, en la posición X del objeto, representado como una variación del parámetro *Loc*.
- *Ciclo 3. Movimiento hacia delante.* En este caso, el sensor recoge el evento cuando se ha pulsado la tecla “flecha hacia delante”, produciendo una variación de 0.01, por cada pulso recibido en el actuador, en la posición del objeto en el eje Y, valor que se selecciona en la segunda columna del parámetro *Loc* del actuador.
- *Ciclo 4. Movimiento hacia atrás.* En este último ciclo, el sensor *keyboard*, recoge el evento de la pulsación de la tecla “flecha hacia atrás”, con lo que varía la posición Y del objeto un -0.01 por cada pulso recibido por el actuador, lo que implica un cambio en la variable *Loc*.

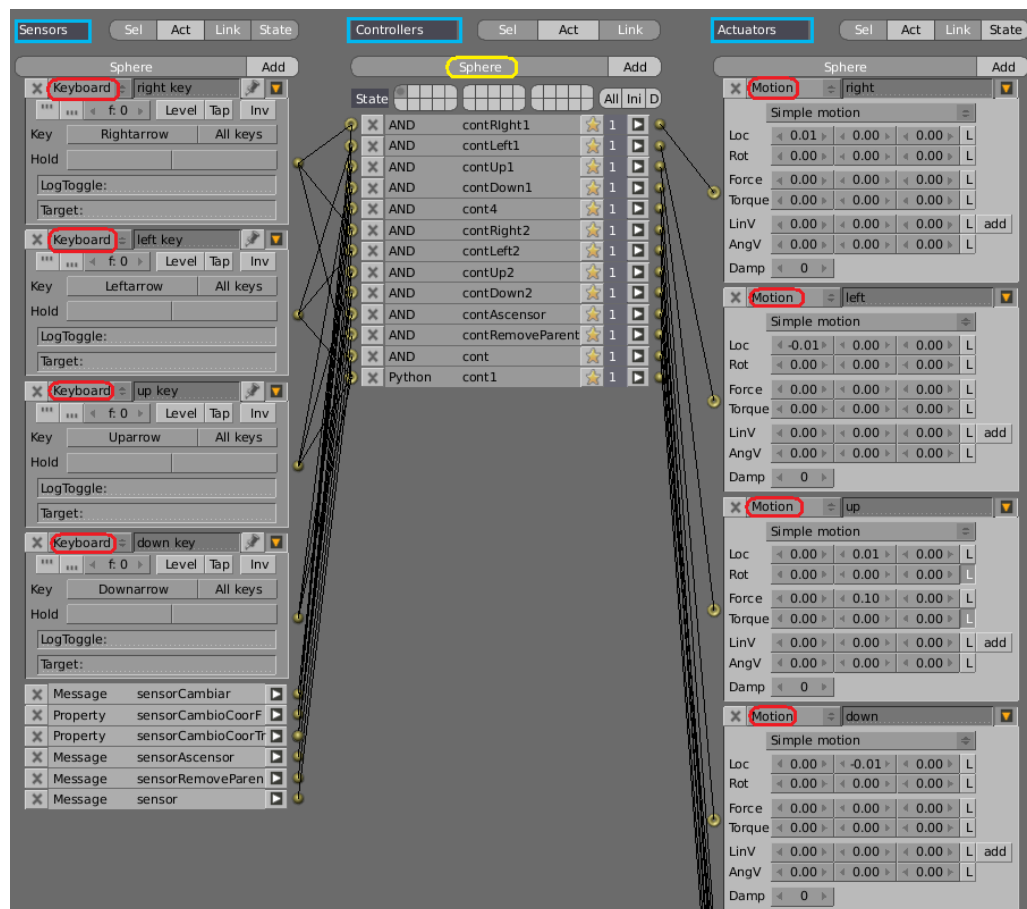


Ilustración 50. Sensors, Controllers & Actuators.

- **Ejemplo 3.**

En este ejemplo se va a proceder a la explicación de la implementación de uno de los objetos puerta. No hay que olvidar que sería una continuación en la implementación expuesta en el primer ejemplo. Como en los ejemplos anteriores, se parte de la imagen que nos muestra el espacio de trabajo y resaltado con el recuadro en rojo el objeto a implementar.

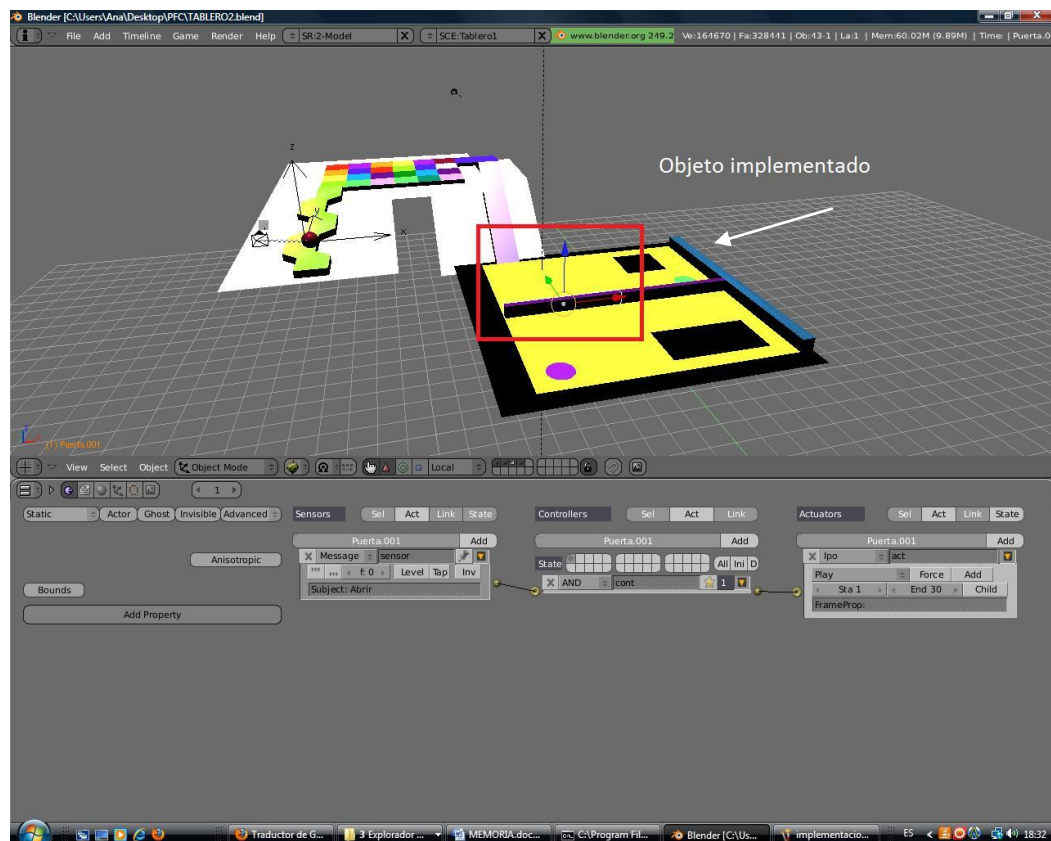


Ilustración 51. Blender workspace ejemplo 3.

Como se aprecia en esta imagen, este objeto no tiene asociada ninguna propiedad por lo que se va a explicar directamente la implementación, que como se aprecia en la ilustración 45 se trata de un único ciclo.

El sensor utilizado es de tipo *message*, en cual, recogerá todos los mensajes de tipo *Abrir*, independientemente del actuador que lo envíe. En el momento en el que lo reciba, enviará un pulso al controlador *AND*, y éste, a su vez, enviará un pulso al actuador. El actuador es de tipo *Ipo*, esto significa que previamente se ha diseñado un movimiento específico para este objeto con el editor de curvas IPO, y que únicamente cuando se reciba el mensaje *Abrir*, se ejecutará dicho movimiento. Aparte de la definición de la curva IPO, debemos configurar unos parámetros adicionales con el fin de seleccionar el fragmento de curva que queremos reproducir, para ello disponemos de tres variables *Sta*,

que define el *frame* de inicio de la animación, en este caso el *frame* 1, la variable *End*, que define el *frame* de finalización, en el cual se ha puesto el valor 30, y por último, la configuración *Play* que determina que la animación únicamente se ejecutará una vez.



Ilustración 52. Sensors, Controller & Actuator.

- **Ejemplo 4.**

En este ejemplo se explica el uso de los *script python* en el *Blender* y la implementación programada en ellos. En la exposición visual se va a ver el *script* que implementa la cuenta atrás del cronómetro. En la imagen que se muestra a continuación, ilustración 46, aparece una tercera ventana en el *workspace*, se trata de un editor de texto donde se pueden implementar los scripts que se deseen.

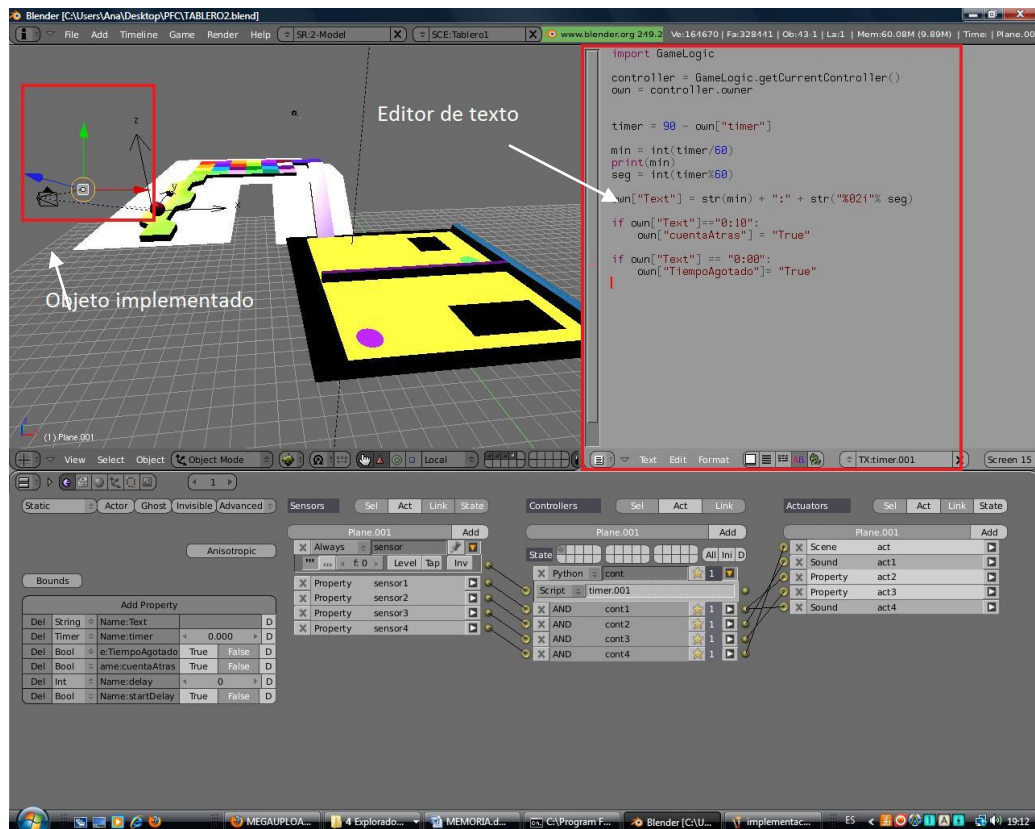


Ilustración 53. Blender workspace ejemplo 4.

El *script*, como ya se ha remarcado, está escrito en *Python*, a continuación se va a explicar, línea a línea, lo implementado en el script:

- **Línea 1.** Importación de la librería *GameLogic*, api que contiene todos métodos que se pueden utilizar para la implementación de la lógica de juegos en *Blender*.
- **Línea 3.** Captura del controlador de *Blender*.
- **Línea 4.** Captura del objeto *own* a partir del controlador, con el fin de capturar la implementación gráfica desarrollada hasta el momento y de este manera asociar nueva variables a nuestro diseño, pero esta vez, de manera textual.
- **Línea 7.** Inicialización de una nueva variable de tipo *Timer*, llamada *timer*, con el valor en el que queremos que empiece la cuenta atrás, en este caso, 90 segundos menos el valor en segundos capturado por el contador interno, *own["timer"]*.
- **Líneas 9 y 10.** Inicialización de una variable de tipo *Integer* que almacena el minuto en el que se encuentra el contador interno, se denomina *min* y se calcula a partir del tiempo propio, calculado internamente, entre 60. La línea diez se corresponde con una traza de seguridad.
- **Línea 11.** Inicialización de una variable de tipo *Integer* que almacena el segundo en el que se encuentra el contador interno, se denomina *seg* y se calcula como el resultado del complementario entre el tiempo actual y el valor 60.
- **Línea 13.** Obtenemos la variable gráfica *Text* de la configuración propia para añadirle el *string* que se mostrará por pantalla con la información del tiempo. La variable *%02i* indica que el valor de la variable *seg* se mostrará siempre en dos dígitos y será de tipo *int*.
- **Líneas 15 y 16.** Se realiza una comprobación, si el *string* de la variable *Text* es igual a "0:10", significa que se inicia la cuenta atrás de los últimos diez segundo y por tanto, se cambia el valor de la variable *cuenta Atrás*, definida gráficamente, a *True*.
- **Líneas 18 y 19.** Se realiza otra comprobación, si el *string* mostrado por la variable *Text* es, en este caso, "0:00", significa que la cuenta atrás se ha agotado, y por tanto, se cambia el valor de la variable *TiempoAgotado* a *True*;

```

1  import GameLogic
2
3  controller = GameLogic.getCurrentController()
4  own = controller.owner
5
6
7  timer = 90 - own["timer"]
8  #print(timer)
9  min = int(timer/60)
10 print(min)
11 seg = int(timer%60)
12 #print(seg)
13 own["Text"] = str(min) + ":" + str("%02i"% seg)
14
15 if own["Text"]=="0:10":
16     own["cuentaAtras"] = "True"
17
18 if own["Text"] == "0:00":
19     own["TiempoAgotado"] = "True"

```

Ilustración 54. Implementación de la cuenta atrás.

Por último, en la ilustración 48, se puede observar cómo se asocia dicho *script* a la ejecución del juego. El sensor utilizado debe ser de tipo *Always* para que la implementación esté activa durante toda la ejecución del juego, y además se debe seleccionar el botón representado con tres puntos suspensivos en la parte superior de éste, para que el script se ejecute cada segundo y por tanto la cuenta atrás vaya cambiando cada segundo. Como se ve, el controlador es de tipo *Python* y en él se indica el nombre del archivo de texto en el que hemos hecho nuestra implementación.



Ilustración 55. Sensors, Controller & Actuators.

• Ejemplo 5.

En este ejemplo se ilustra el uso del actuador de tipo *scene*. Como se ha dicho en la explicación de los diferentes tipos de actuadores, apartado 5.3.6, se ha usado este actuador para dos conceptos diferentes, uno para cambiar las cámaras dentro de cada nivel y otro, el que se explica en este ejemplo, para cambiar de una escena a otra, puesto que cada nivel se ha diseñado en una escena diferente. El cambio de escena puede hacerse dentro de una misma animación o para pasar de una animación a otra de manera simple o incluyéndole una transición. En este caso que nos ocupa, el cambio de escena

se produce cuando el jugador ha conseguido superar el primer nivel, por tanto, se produce un cambio de escena sin transiciones entre el primer nivel y el comienzo del segundo nivel. En la ilustración 49, se remarca con un recuadro de color rojo el objeto al que se le va a aplicar el actuador de cambio de escena, y con un recuadro verde, la escena en la que nos encontramos *Scene1*.

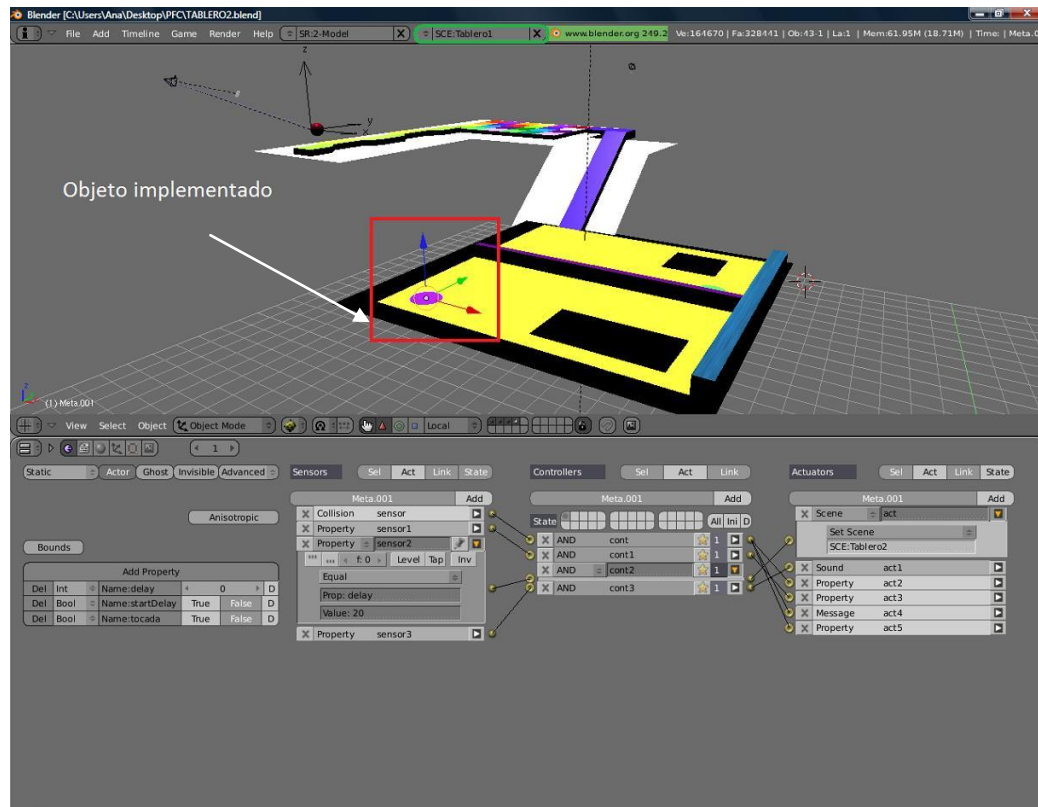


Ilustración 56. Blender workspace ejemplo 5.

Como se aprecia en la ilustración 50, se han definido tres variables para este objeto, la única que es de interés para esta explicación es la variable *delay* de tipo *Integer* e inicializada a valor cero. Para el cambio de escena basta con un ciclo de implementación simple. En este caso, se cuenta con un sensor de tipo *property* que envía un pulso cuando la variable *delay* tenga valor 20, este pulso será recibido por un controlador *AND* y este a su vez le enviará un pulso al actuador de tipo *scene*, donde se le indicará en la configuración “*Set Scene*”, seleccione escena, la escena a introducir, en este caso, “*SCE: Tablero2*”.

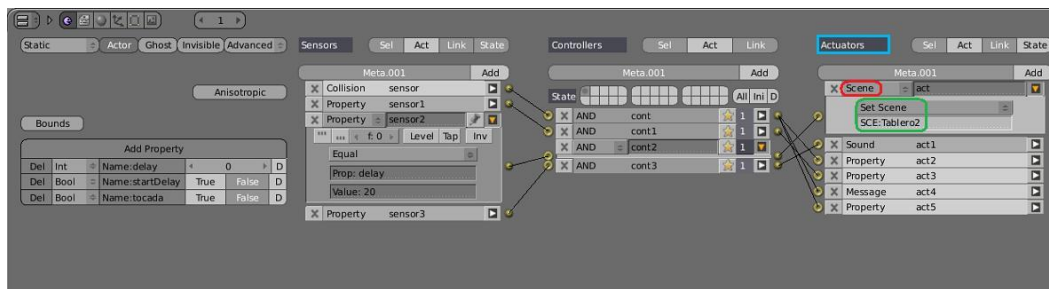


Ilustración 57. Implementación actuador Scene.

• Ejemplo 6.

En el ejemplo que se ilustra en las imágenes número 51 y 52, se va a explicar el uso del actuador *parent*. Este actuador es muy útil cuando se quiere que una acción se aplique a más de un objeto a la vez y que los movimientos de uno sean repetidos por uno o varios objetos. En este caso, el actuador *parent* se usa para que cuando el objeto esférico colisione con el plano circular de color morado, la esfera se quede inmóvil encima de la meta, suprimiendo la interacción del usuario con el personaje.

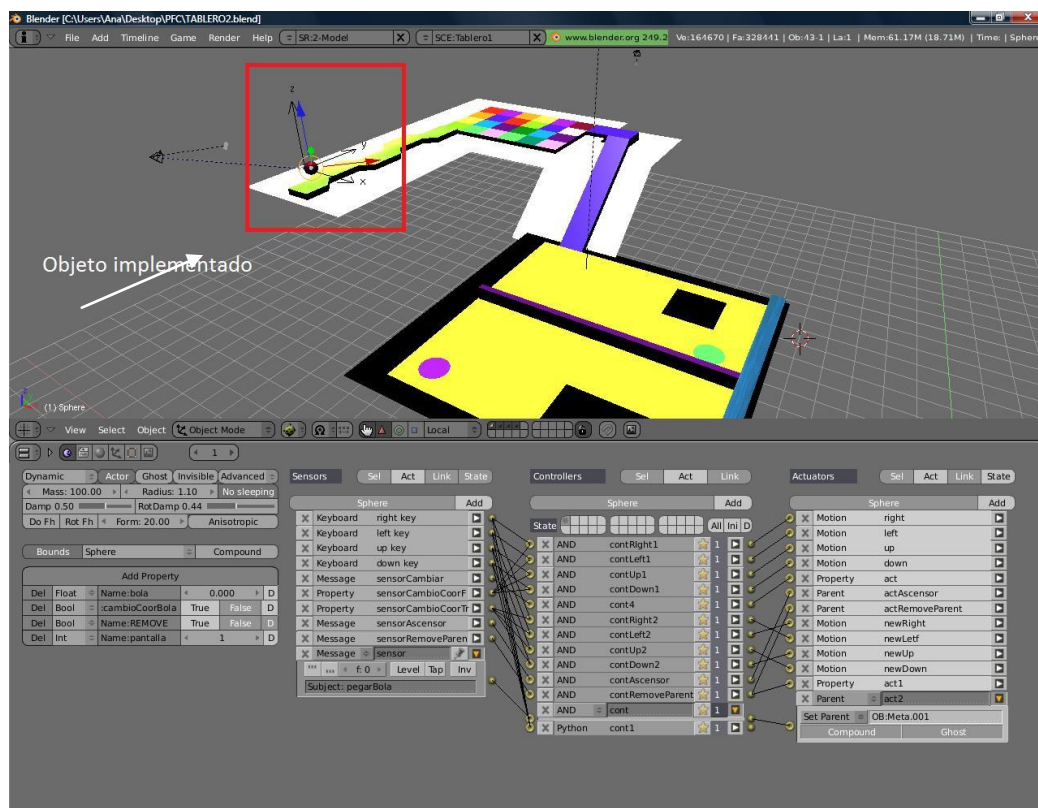


Ilustración 58. Blender workspace ejemplo 6.

Por tanto, se trata de un ciclo de implementación simple, donde un sensor de tipo *message*, enviará un pulso cuando reciba el mensaje “pegarBola”, para luego ser recibido por un controlador de tipo *AND*, que enviará un pulso al actuador de tipo *parent*, donde se selecciona el objeto que se convierte en el padre, en este caso el objeto “Meta.001”, en la configuración “Set Parent”.

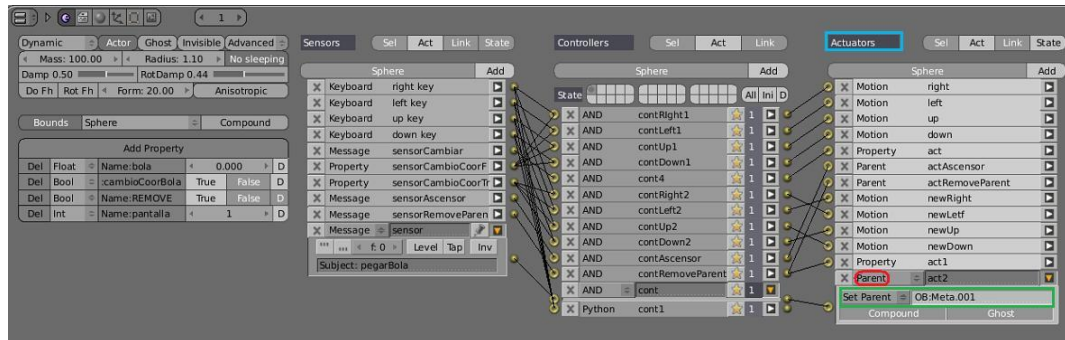
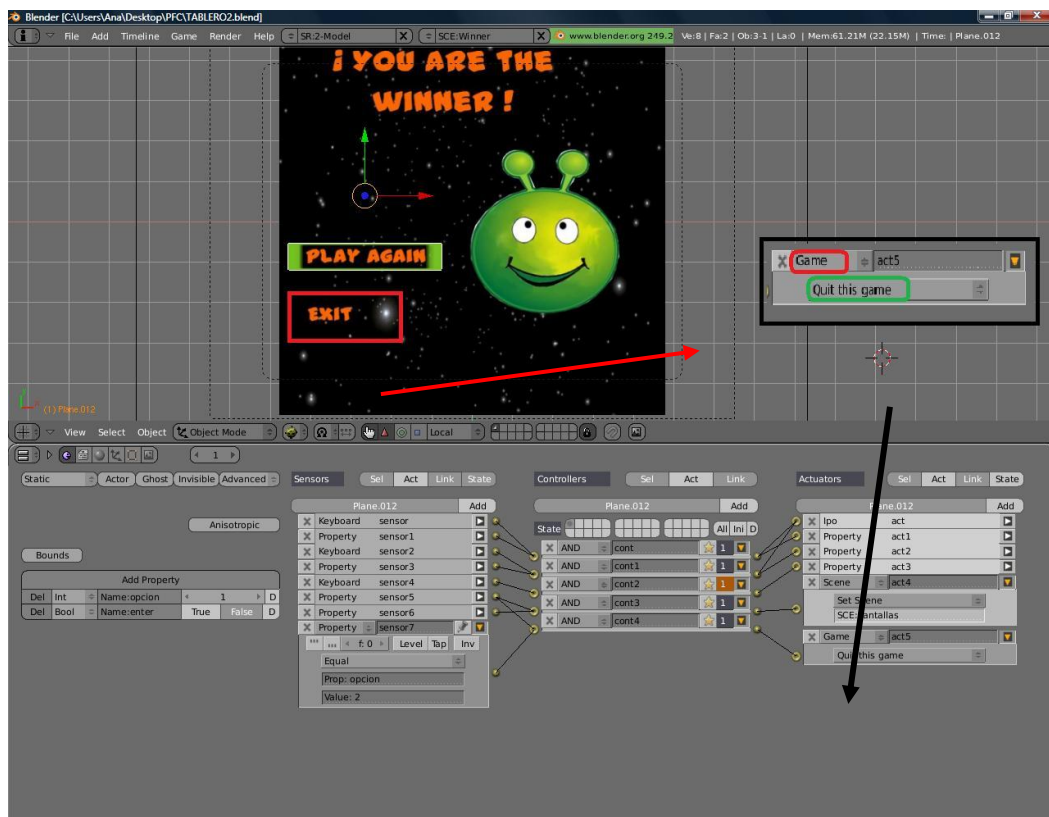


Ilustración 59. Implementación *actuador parent*.

• Ejemplo 7.

En este último ejemplo se va a explicar el uso del actuador *Game*. Este actuador se utiliza únicamente en la última pantalla, que sólo se ejecuta cuando el jugador ha conseguido superar los tres niveles propuestos. Esta pantalla forma parte de la interfaz gráfica de usuario del juego, y en ella el usuario deberá elegir entre dos opciones, volver a jugar o salir del juego.

Ésta última, *exit*, ejecuta el siguiente ciclo de implementación. La opción *exit* del menú está diferenciada a partir del valor de la variable *opción*, asignada al objeto que se utiliza para la selección del menú. Por tanto, mediante el sensor *property* se evalúa si la variable *opción* tiene el valor numérico 2, si es así, se envía el pulso al controlador y este al actuador de tipo *Game*, donde se selecciona la opción “quit this game”, forzando así la finalización de la ejecución del juego.

Ilustración 60. Implementación *actuador game*.

Como se ha podido apreciar, casi todos los ejemplos explicados han sido sacados del primer nivel, esto ha sido para facilitar la explicación de los ejemplos y hacer comprender que el sistema de implementación es prácticamente similar para cualquier cosa que se quiera realizar, con la única diferencia de tener que manejar un mayor número de variables.

4.4. POST-PRODUCCIÓN.

La post-producción en un videojuego es casi tan importante como lo puede ser en una película, el desarrollo no termina con la implementación y diseño de los niveles, sino con la unión de los niveles dentro de un flujo donde la interfaz de usuario es esencial, para que el usuario controle la línea temporal del videojuego. Además, hay que añadirle los efectos de sonido y generar un archivo ejecutable que contendrá todos los ficheros necesarios para la ejecución del videojuego en cualquier máquina.

4.4.1. Efectos de sonido.

Como ya hemos explicado en el apartado de sonido (5.3.2), los efectos son primordiales en la post-producción para obtener el resultado final que se quiere conseguir.

Para integrar los sonidos descritos en ese mismo apartado, se deberá utilizar uno de los actuadores descritos en el apartado de diseño de programación (5.3.6) y que a su vez está explicado en el ejemplo 2 del mismo apartado.

4.4.2. Interfaz gráfica de usuario.

La interfaz gráfica de usuario está compuesta por una serie de ventanas que hacen la función de menú a través de las cuales el usuario puede ir eligiendo las acciones que desea hacer.

Como se puede observar en la ilustración 54, el menú principal dispone de tres opciones entre las que el usuario debe elegir:

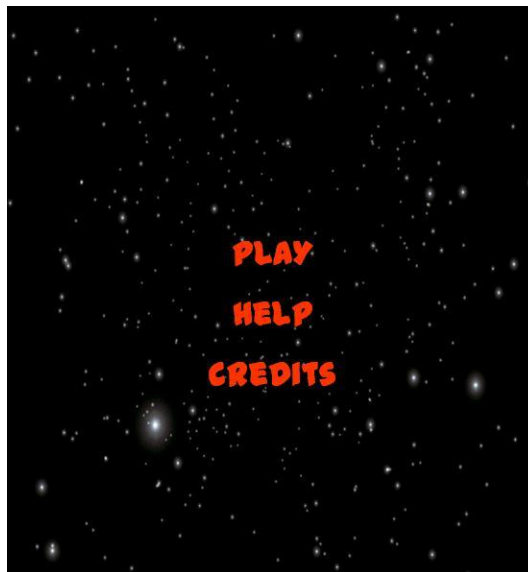


Ilustración 61. Ventana MENÚ PRINCIPAL.

- **Play.** Esta opción llevará al usuario a otra ventana donde podrá elegir entre los tres niveles disponibles.
- **Help.** Esta opción llevará al usuario a otra ventana donde encontrará una pequeña ayuda para entender la mecánica del juego.
- **Credits.** Esta opción llevará al usuario a otra ventana donde encontrará información sobre los desarrolladores del juego.

La ventana que aparecerá cuando el usuario pulse la opción “*play*” se ilustra en la imagen 55, en ella aparecen tres imágenes que representan los tres niveles disponibles, cuando el usuario elija alguno de ellos automáticamente ese nivel se ejecutará y comenzará el juego.

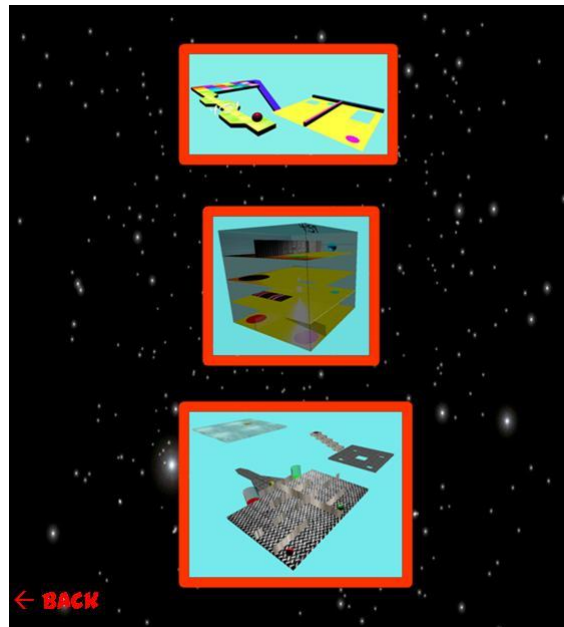


Ilustración 62. Ventana ELEGIR NIVEL.

La ventana que aparecerá si el usuario pulsa la opción “*help*” será la ilustrada en la imagen 56, y en el caso en el que pulse la tecla “flecha hacia la derecha” verá la ventana que se representa en la ilustración 57. El contenido de estas dos ventanas explica brevemente los controles que el usuario debe utilizar para moverse durante el juego, el significado de los círculos para que sepa que acciones realizan en función del color y por último, muestra una pequeña advertencia para que sepan que existe un cambio de gravedad en algún momento de la partida. Como ya se ha dicho al principio de este párrafo, la navegación entre las dos páginas se hace mediante la flecha y viene indicado en las esquinas inferiores de las ventanas.



Ilustración 63. Ventana AYUDA I.



Ilustración 64. Ventana AYUDA II.

En el caso en el que el usuario pulse la opción “credits” del menú principal, le aparecerá la ventana ilustrada en la imagen 58. Esta ventana, únicamente, ofrece información sobre los desarrolladores.



Ilustración 65. Ventana CRÉDITOS.

Además de estas ventanas de navegación, hay dos ventanas más que aparece en el transcurso del juego. Una será visible en el caso en el que el usuario falla y la otra cuando el usuario ha conseguido completar todos niveles.

La ventana *Game Over*, representada en la ilustración 59, tiene tres opciones posibles:

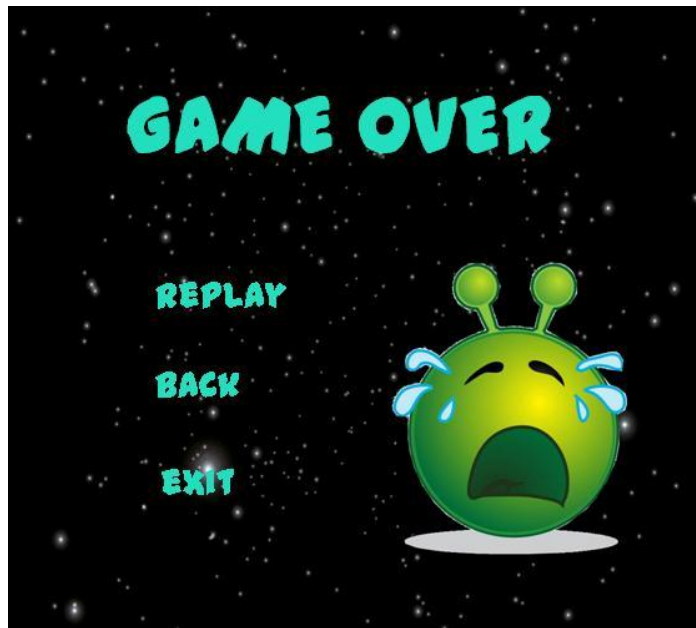


Ilustración 66. Ventana GAME OVER.

- **Replay.** Si el usuario elije esta opción automáticamente se reiniciará el nivel donde ha fallado.
- **Back.** Para volver al menú principal.
- **Exit.** Para terminar la ejecución del videojuego.

La ventana *Winner*, ilustrada en la imagen 60, tiene dos opciones:



Ilustración 67. Ventana WINNER.

- **Play again.** Si el usuario elige esta opción, automáticamente le llevará a la ventana donde se eligen los niveles.
- **Exit.** Para terminar la ejecución del juego.

4.4.3. Generación del archivo ejecutable.

Generar el archivo ejecutable es muy sencillo, en la ventana principal de Blender, en la barra de herramientas hay una opción que se llama *File*, en su menú desplegable hay una opción que se denomina “*save as runtime*”. A continuación, aparece seleccionado el directorio en el que se guardarán los archivos *.blend* y es ahí donde se creará un archivo ejecutable con el nombre que se desee, en este caso *GAME.exe*, y los archivos *.dll* necesarios para la ejecución del videojuego.

Por tanto, los usuarios que quieran disponer del juego deben tener una carpeta que contenga dichos archivos. Además, de estos, debe incluir las carpetas con las imágenes, texturas y archivos *.mp3*, que están incluidos en el videojuego.

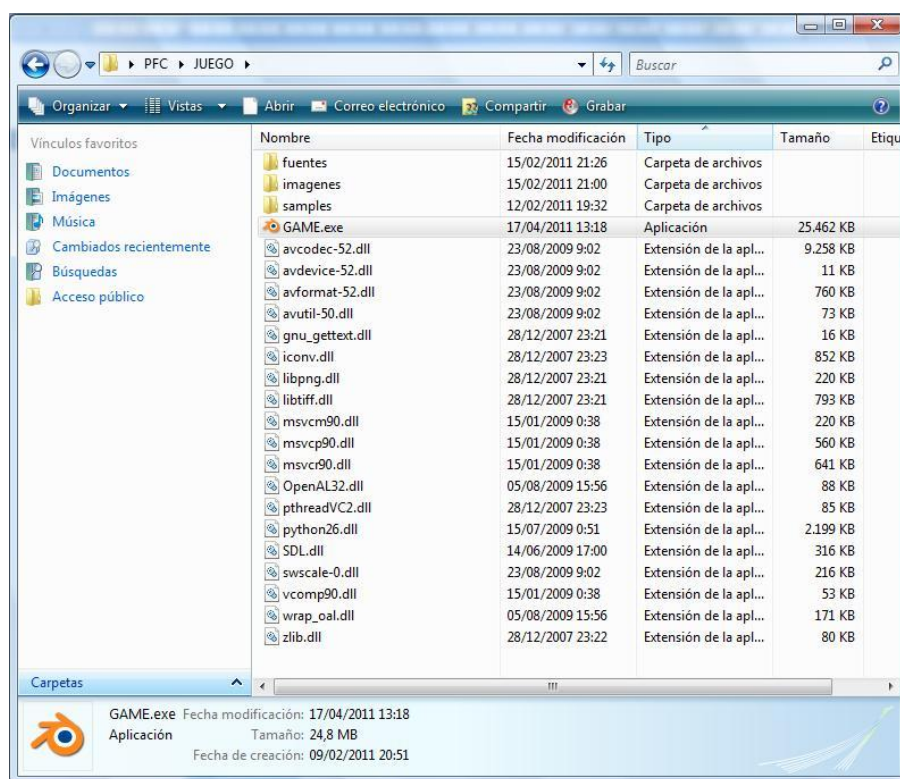


Ilustración 68. Archivos necesarios para la ejecución del juego.

5. FASE DE TESTING

En toda creación de un proyecto se realiza una fase de prueba para evaluar el funcionamiento, con el fin de detectar posibles errores para poder resolver y también, para determinar los puntos fuertes y débiles de cara al público. Esta fase es muy importante en el desarrollo de un videojuego, ya que la opinión de los usuarios es vital para su éxito. No sólo se evalúa que la implementación sea buena, sino la repercusión que tiene en el jugador.

Se deben medir muchos aspectos para evaluar un videojuego, pero hay dos que son especialmente importantes y vitales, para que el impacto en los jugadores sea el que estábamos buscando, estos son la jugabilidad y la usabilidad.

5.1. LA JUGABILIDAD

Entre los test que se realizan a un videojuego antes de su salida al mercado, se realiza uno en particular que mide principalmente la “jugabilidad”. Este término que no está reconocido por la Real Academia de la Lengua Española como tal, sino que se trata de una traducción directa de la palabra anglosajona “*playability*”, nació en los años 80. Se entiende por jugabilidad, como el conjunto de propiedades que describen la experiencia del jugador ante un sistema de juego determinado, cuyo principal objetivo es divertir y entretener de forma satisfactoria y creíble, ya sea solo o en compañía de otras personas.⁶

José Luis González Sánchez, definió jugabilidad en su tesis doctoral como “término empleado en el diseño y análisis de juegos que describe la calidad del juego en términos de sus reglas de funcionamiento y de su diseño como juego. Se refiere a todas las experiencias de un jugador durante la interacción con sistemas de juego”⁷

5.2. JUGABILIDAD vs USABILIDAD

En multitud de ocasiones se confunden el concepto de usabilidad y el de jugabilidad, debido a su significado en el ámbito en el que nos encontramos es muy similar. Sin embargo, son conceptos diferentes y ambos son muy importantes para el análisis de un videojuego.

Usabilidad en el campo del desarrollo de videojuegos se entiende como un concepto inherente a la *HCI (Human Computer Interaction)*, y también se trata de una traducción literal del término anglosajón “*Usability*”, que significa facilidad de uso. Se

⁶ Referencias 1

⁷ Referencias 2

define como el grado de eficacia, eficiencia y satisfacción con la que los usuarios pueden lograr objetivos específicos, en contextos de uso determinados.⁸

Por tanto, podemos extraer información muy interesante analizando la experiencia desde el enfoque de la usabilidad y desde la jugabilidad.

A pesar de ser dos conceptos que vienen a ser similares en muchos aspectos, tienen características que los diferencian. Son éstas las que debemos tener en cuenta para el diseño de nuestro videojuego. Para poder entender en un principio ambos, podemos enfocar la usabilidad hacia la manera en la que los jugadores se enfrenta al videojuego en el concepto de aplicación y la jugabilidad al entretenimiento que experimenta con él.

A continuación se muestra una tabla con los objetivos que hay que cumplir en ambos conceptos a la hora del diseño⁹:

Objetivos de Usabilidad	Objetivos de Jugabilidad
Realizar una tarea eficiente	Entretener el máximo tiempo posible
Eliminación de posibles errores	Divertir superando obstáculos
Recompensa externa: Trabajo realizado	Recompensa interna: Diversión
Facilidad de aprendizaje e intuición	Nuevas cosas por aprender y descubrir
Reducción de la carga de trabajo	Incrementar mecánicas y metas de juego
Asume que la tecnología debe ser humanizada	Asume que el jugador debe ser retado por la tecnología

Tabla 14. Objetivos usabilidad y jugabilidad.

5.3. PROPIEDADES DE LA JUGABILIDAD

Para medir la jugabilidad de un videojuego, es necesario definir varios factores. Su estudio nos proporcionará información muy válida para evaluar un proyecto. Hay que relacionar todos los factores entre sí para obtener un buen resultado del análisis.

Partiendo de la tesis doctoral realizada por José Luis González Sánchez para la Universidad de Granada, podemos diferenciar varios factores que definen más exhaustivamente la experiencia del jugador.

5.3.1. Satisfacción.

Se define satisfacción como el agrado o complacencia que el jugador ha experimentado una vez completado todo el juego. Es el atributo más importante, ya que en último término, un juego tiene éxito si es capaz de proporcionar diversión al mayor número de personas. Por eso, el fin del desarrollador es que su videojuego divierta y entretenga al usuario.

⁸ Referencias 3

⁹ Referencias 2

Tiene un alto nivel de subjetividad, por lo que su medición es bastante compleja. Por eso para evaluarlo es necesario tener claro varios términos. Uno de ellos es el tipo de público al que va dirigido el juego, pues no es lo mismo la satisfacción que experimenta un niño con un juego adecuado para su edad, como puede ser la saga “Rayman”, que la que experimenta un adulto con la saga “Grand Theft Auto”. Otro es la plataforma para la que están desarrollados los videojuegos, ya que está comprobado que las consolas de nueva generación tienen varios elementos que aumentan la satisfacción del usuario, como pueden ser accesorios (pistolas, volantes,...) adicionales.

La satisfacción se caracteriza por varias propiedades:

- **Diversión.** Es fundamental para la satisfacción del usuario que el videojuego sea divertido para él, por eso es una de las propiedades más importantes. Se trata de un indicador muy subjetivo. Una de las maneras de analizarlo es ver cuánto tiempo permanece jugando el usuario o el porcentaje de juego que ha descubierto.
- **Placer.** El desarrollador debe impedir que el usuario sufra demasiada frustración durante el juego. Esto supondría una bajada muy importante del grado de interés, por lo que la satisfacción del jugador iría en decremento. Esta propiedad está muy ligada al tiempo de aprendizaje del juego, a mayor tiempo de aprendizaje, mayor frustración y menor placer. Por eso, un videojuego debe ser intuitivo en su mecánica (*Game Mechanic*), es decir debe proporcionar la ayuda necesaria para que el desarrollo del juego sea sencillo, pero a la vez no excesiva para añadirle un sentimiento de inquietud que aumente el interés.
- **Atractividad.** Un videojuego debe ser atractivo para el usuario. No solo mediante elementos visuales, sino a través del desarrollo argumental de la historia, la mecánica, la interactividad y los efectos sonoros.

5.3.2. Aprendizaje.

Facilidad para comprender y dominar el sistema y la mecánica del videojuego, es decir, los conceptos definidos en el *Gameplay/Game Mechanic* del juego: objetivos, reglas y formas de interaccionar con el videojuego¹⁰.

¹⁰ Referencias 2.

Cuando se habla de aprendizaje, nos centramos en que las metas y objetivos del juego sean fáciles de comprender, así aumentamos la interacción del usuario con el juego. Se puede establecer una curva de aprendizaje que representaría el conocimiento que el jugador va adquiriendo en función del tiempo que lleva jugando.

Mediante el análisis de dicha curva se establecen dos fases. La primera fase llamada de adaptación, que finaliza cuando el jugador ya sabe completamente la mecánica del juego siendo capaz de resolver de manera intuitiva los retos que se le vayan imponiendo. Y la segunda fase, denominada fase de estabilidad. Esta fase puede ser peligrosa, ya que puede generar que el jugador entre en una rutina no deseable y acabe por abandonar el juego, sencillamente, porque se aburra de realizar siempre las mismas funciones. Aquí es donde la creatividad del desarrollador, incluyendo nuevos retos, puede hacer que la fase de estabilidad no necesariamente tenga que finalizar con que el usuario abandone el juego.

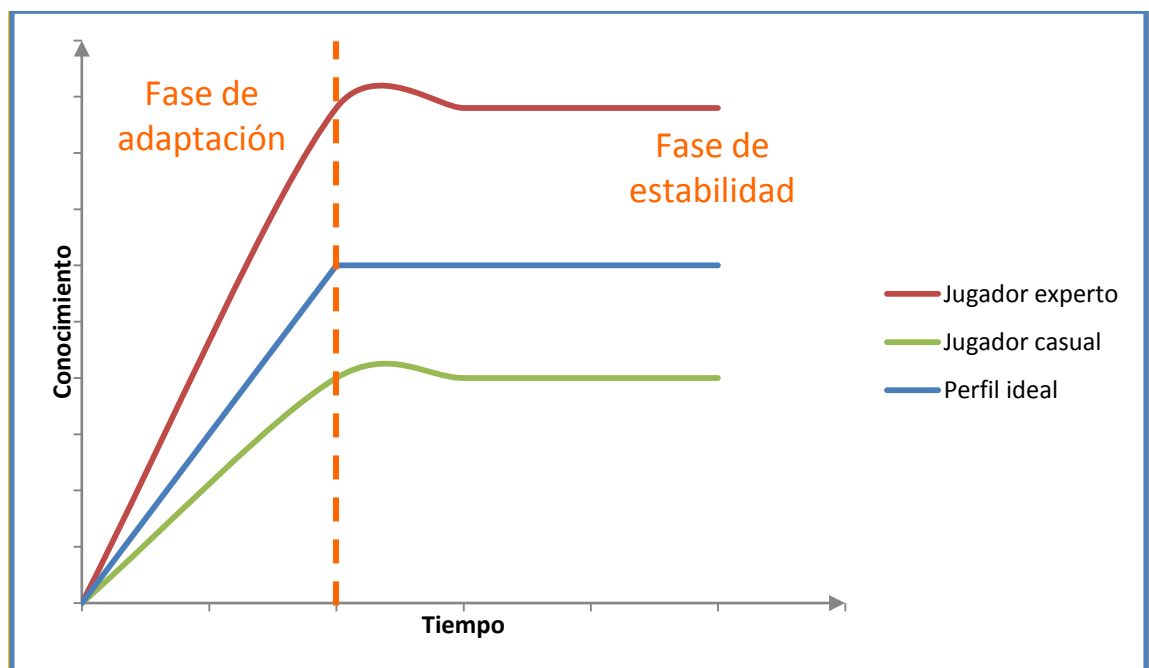


Ilustración 69. Curva de aprendizaje.

Las propiedades que analizan el aprendizaje son:

- **Conocimiento del juego.** El jugador debe conocer la mecánica del juego en un tiempo prudencial. El conocimiento previo de un juego del mismo tipo hace que la fase de adaptación de este jugador sea menor que el de aquél que nunca haya jugado a nada similar. Sin embargo, el hecho de

enfrentarse a una mecánica desconocida generalmente aumenta el interés del jugador.

- **Habilidad del jugador.** Una vez comprendida la mecánica, el jugador debe interactuar con el juego. Su habilidad para resolver los retos que se va encontrando aumentará el grado de aprendizaje.
- **Dificultad.** El grado de dificultad asociado a los objetivos influirá muchísimo en el aprendizaje del jugador. Adaptar el grado de dificultad en cada momento del juego es muy importante para incentivar la implicación del jugador sin llegar a causarle una frustración que haga que abandone el juego. La adaptación del grado de dificultad se realiza generalmente mediante un mayor volumen de ayuda a comienzo del juego, para disminuirlo progresivamente según vaya superando la curva de aprendizaje. De ahí, que éste sea otro factor importante en la medida del aprendizaje.
- **Frustración.** Sentimiento que un jugador sufre cuando no es capaz de solventar el reto que se le ha impuesto. El grado de frustración debe estar muy controlado para no superar el nivel que suponga el abandono de la actividad por parte del usuario, al sentirse incapaz de superar el reto propuesto.
- **Velocidad de los elementos del juego.** Hay que ir introduciendo la mecánica del juego a una velocidad adecuada para que el jugador sienta que está aprendiéndola a un ritmo que ni resulte aburrido ni excesivo. Si este concepto está bien introducido, el jugador irá aprendiendo sin tener conciencia de ello y se irá introduciendo cada vez más en el desarrollo argumental del videojuego.
- **Descubrimiento de la mecánica.** Se debe facilitar al jugador el conocimiento del juego no sólo mediante su propia experiencia y habilidad, sino mediante tutoriales o niveles de entrenamiento previos al juego, para inyectar la confianza suficiente para comenzar la misión que le tengamos encomendada.

5.3.3. Efectividad.

Se trata del tiempo y de los recursos que se le van proporcionando al jugador para aumentar su grado de diversión mientras va logrando los objetivos que se

va encontrando durante el transcurso del juego y que harán que llegue al final de éste.

La efectividad muestra el grado de captación que el juego consigue del jugador. Por tanto, un juego eficiente es aquel que es capaz de mantener la atención del jugador, no solo al principio, sino que mantiene el interés por seguir jugando.

Las propiedades asociadas a este factor son:

- **Complejidad.** Si un jugador ha completado un videojuego en su totalidad, es porque éste ha sido claramente eficiente, suscitando un interés durante todo su desarrollo.
- **Estructuración.** Un juego se considera bien estructurado, cuando sus elementos están bien equilibrados, es decir cuando le vamos ofreciendo gradualmente al jugador las herramientas necesarias para conseguir los objetivos, esto hará que sea eficiente.

5.3.4. Inmersión.

Capacidad para que el usuario empatice con lo que está jugando, de tal manera que pueda integrarse en el mundo virtual que se muestra. Es muy importante hacer que el jugador forme parte del juego y forme parte como protagonista.

Su medida depende de estas propiedades:

- **Conciencia.** Cómo de creíble lo percibe el jugador, tanto en la percepción visual como en los sentimientos que le producen. Esto producirá que la implicación del jugador sea mayor. La conciencia que éste adopte frente al juego le influirá a la hora de afrontar los retos.
- **Concentración.** Es importante hacer que el jugador mantenga la concentración en el juego para que esté perfectamente integrado dentro del mundo virtual diseñado para él.
- **Realismo.** Si los movimientos efectuados a través de los controles dados, los objetivos y la ambientación, hacen que el jugador se sitúe en un mundo realista, su grado de inmersión será superior,

proporcionándole entonces una sensación similar a la que obtendría si la acción la estuviera realizando él mismo, ayudándolo a superar los retos.

- **Destreza.** Se trata de la habilidad que tiene el jugador para ir consiguiendo los objetivos. Es un factor que no es posible controlar, sin embargo, es muy importante en el grado de inmersión del juego.
- **Cercanía sociocultural.** A la hora de desarrollar un videojuego, es muy importante saber al grupo social al que va destinado. Si nos equivocamos en este sentido la inmersión de los jugadores que prueben nuestro videojuego será nula y nuestro estudio inútil.

5.3.5. Motivación.

Los videojuegos deben someter al usuario a diversos retos, mediante distintas herramientas, con la intención de conseguir un reto claro. Para poder conseguir que esta función no se convierta en una obligación el desarrollador debe infligir una motivación al jugador.

Las propiedades que la componen son:

- **Estimulación.** Es muy importante la estimulación que se genere en el jugador para afrontar los retos que se le imponen.
- **Curiosidad.** Si el juego consigue despertar la curiosidad del usuario, crecerá su motivación.
- **Automejora.** Puede venir dada por nuevas habilidades adquiridas a lo largo del juego.
- **Diversidad.** Que los retos vayan variando hace que el juego sea menos monótono y aumenta la motivación del jugador.

5.3.6. Emoción.

Es un impulso involuntario que se produce como respuesta a los estímulos que genera el videojuego en el jugador y que desencadena emociones

espontáneas. Es la esencia de un videojuego, pues éste tiene que ser capaz de generar estímulos que aumenten el interés por seguir jugando. Un juego debe ser capaz de generar miedo, alegría y frustración, casi al mismo tiempo, no solo por los retos que debe afrontar el jugador, sino también por su apariencia visual y sus efectos sonoros.

La emoción se caracteriza por las siguientes propiedades:

- **Reacción.** La reacción del jugador a los estímulos que recibe por parte del videojuego depende de su desarrollo emocional. A mayor reacción con los estímulos mayor grado de emoción.
- **Conducta.** Los videojuegos producen que el jugador vaya cambiando su conducta durante todo el desarrollo de la historia. Cuando el videojuego es capaz de ir cambiando la conducta del jugador, mayor grado de emoción le produce.
- **Atracción sensorial.** Esta propiedad está ligada a los canales de estimulación que el juego utiliza. A mayor número de canales, mayor grado de implicación del usuario. Éstos le ayudan a formar parte del juego, generando las emociones que el desarrollador quiere proporcionarle.

5.3.7. Socialización.

Atributos y elementos del juego que fomentan el factor social o la experiencia en grupo, lo cual provoca apreciar el videojuego de distinta manera, gracias a las relaciones que se entablan con el resto de jugadores. Son muy diferentes las emociones que siente un jugador cuando juegue solo o integrado en un grupo. Este factor puede hacer que los anteriormente descritos se vean potenciados enormemente. Un grupo incentiva la competencia y esto implica el aumento de la alegría o de la frustración. En juegos con un único jugador, la autocompetencia se consigue a través de diferentes retos, uno de ellos, por ejemplo, el hecho de tener que conseguir el objetivo en un tiempo dado.

Las propiedades asociadas a este factor son:

- **Percepción social.** A menudo la percepción de un juego varía en función de la posibilidad de que haya varios jugadores en lugar de un único jugador.
- **Conciencia de grupo.** El videojuego tiene que generar una conciencia de grupo para que este consiga los objetivos que se plantean. Solo obtendrán el triunfo si trabajan en equipo.
- **Implicación.** La implicación que el videojuego genere en el grupo es muy importante para el desarrollo de los retos.
- **Compartición.** El grupo debe compartir los recursos que facilita el juego.
- **Comunicación.** Un videojuego en grupo requiere que sus miembros, se comuniquen si desean conseguir el objetivo. Por tanto, es un factor muy importante para la socialización.
- **Interacción** de todo el grupo con el videojuego.

Todos estos factores, deben ser medidos para analizar cómo los jugadores reaccionan ante nuestra propuesta de videojuego. Es muy importante para determinar en qué aspectos falla nuestro proyecto, para poder mejorarlo, y cuáles son buenos, para potenciarlos.

5.4. FACETAS DE LA JUGABILIDAD

Se puede entender la jugabilidad dentro de muchos aspectos y por tanto se da la posibilidad de diferentes formas de interpretarla. Las facetas de la jugabilidad nos permiten entender mejor los factores descritos en el punto anterior y determinar los distintos puntos de vista que este concepto nos presenta.

- **Jugabilidad intrínseca.** Se mide con referencia a la propia naturaleza del juego y de cómo los desarrolladores la han proyectado hacia el jugador. Está ligada al *GamePlaying* y al *Game Mechanics*, teniendo en cuenta además la implementación que los desarrolladores han realizado para determinar estos dos aspectos. En esta faceta se analiza la representación de las reglas, los objetivos a conseguir, el ritmo establecido en el juego y la misma mecánica de este.

- **Jugabilidad mecánica.** Se trata de la jugabilidad asociada a la calidad software del videojuego, por tanto está ligada al Game Engine. En ella se mide la fluidez de las escenas que contienen movimiento, la correcta iluminación de las escenas, los efectos sonoros, el comportamiento de los personajes en cada momento del juego y por supuesto, los sistemas de comunicación en juegos multijugador que requieren de una correcta sincronización a pesar de las fluctuaciones en la red.
- **Jugabilidad interactiva.** Se basa en la usabilidad de la interfaz gráfica de usuario disponible en el juego, así como los mecanismos de diálogo y los sistemas de control.
- **Jugabilidad artística.** Asociada a el aspecto estético del videojuego. Un juego puede tener una jugabilidad intrínseca impecable, pero si su aspecto visual no es bueno, puede disminuir mucho su calidad y su análisis de la jugabilidad respecto a los usuarios puede no ser la deseable. Los elementos que determinan esta faceta son la calidad gráfica, los efectos sonoros, el argumento y narración de la historia y la integración de todos ellos en el videojuego.
- **Jugabilidad intrapersonal.** Expresa únicamente la percepción que el usuario tiene del juego, por tanto, es una faceta totalmente subjetiva.
- **Jugabilidad interpersonal.** Muestra la percepción que tienen un grupo de jugadores cuando juegan juntos, la competitividad, la colaboración entre ellos y la satisfacción que obtienen.

5.5. MI PROPIA FASE DE TESTING

En este apartado se van a explicar los dos tipos de test realizados así como sus resultados. La fase de testing nos muestra aspectos que desde el punto de vista del desarrollador seríamos incapaces de apreciar.

Para realizar un mejor estudio del desarrollo hemos dividido la fase de testing en dos partes:

- **Test de prueba del videojuego.** Mediante este test vemos los diferentes errores que se producen en la ejecución del videojuego, para luego analizarlo y buscar las posibles soluciones.
- **Test de jugabilidad.** Mediante este test recogemos todas las características explicadas en los apartados anteriores. De esta manera, vemos cuales son los

puntos débiles de nuestro videojuego, para así realizar los cambios oportunos para que se conviertan en puntos fuertes.

Para la realización de la fase de testing hemos contado con la colaboración de un grupo de personas voluntarias.

5.5.1. Test de pruebas.

El test de pruebas se realiza con el fin de encontrar posibles errores durante la ejecución del videojuego. Estos errores sólo son visibles una vez se ha terminado el desarrollo. Son errores que pasan inadvertidos a la vista del desarrollador, esto se debe, a que son muy concretos ya que depende de un caso determinado dentro de la ejecución.

Como se ha dicho anteriormente, contamos con un grupo de personas que son las que van a realizar este test voluntariamente. Cada una de ellas ha jugado una partida completa mientras era supervisado por uno de los desarrolladores. De esta manera, en el momento en el que el desarrollador detecta un error de ejecución lo anota para su posterior estudio.

Los errores encontrados durante esta fase y sus posibles soluciones vienen descritos en la siguiente tabla. En ella podemos ver:

- Una descripción breve del error.
- La frecuencia con la que se produce.
- El grado de importancia del error, es decir, si se trata de un error en el que hay que forzar la finalización de la ejecución porque el usuario ya no puede continuar con el nivel o si se trata de un error que permite que el juego se siga ejecutando. Para ello tenemos dos tipos:
 - E (*Error*), error grave, es necesario terminar con la ejecución del videojuego, porque el flujo del programa se ha roto.
 - W (*warning*), se trata de una advertencia, es decir, es un error que convive perfectamente con el flujo de ejecución.
- Solución adoptada.

ERROR 1: Encontrado en la pantalla 2.

Descripción	Frecuencia	Importancia	Posible solución
Una vez que la bola ha cambiado de tamaño y que cae al último nivel del cubo, el plano que representa al suelo en este nivel no detecta la bola y se sale del cubo.	Ocasionalmente	E	Aumentar el parámetro que representa al volumen de detección de la bola, para que aunque sea más grande aunque visualmente sea más pequeña, de esta manera el plano será capaz de detectar la colisión de la bola cuando caiga encima de él.

Tabla 15. ERROR 1.

ERROR 2: Encontrado en la pantalla 2.

Descripción	Frecuencia	Importancia	Posible solución
Una vez que la bola ha cambiado de tamaño y toca la cinta transportadora, esta debería expulsarla del cubo y por lo tanto el jugador debería empezar de nuevo, sin embargo no lo hace.	En raras ocasiones	W	Añadir una relación de parentesco radical para que la bola se quede “pegada” en el momento en el que toque la cinta.

Tabla 16. ERROR 2.

ERROR 3: Encontrado en la pantalla 3.

Descripción	Frecuencia	Importancia	Posible solución
Para la detección de la caída de la bola, colocamos un plano justo debajo de la plataforma. Ocurría que una de las dos bolas se salía del tablero pero no se detectaba.	En raras ocasiones	E	Aumentar el plano que detecta las caídas en ambos ejes (X e Y).

Tabla 17. ERROR 3.

ERROR 4: Encontrado en la pantalla 3.

Descripción	Frecuencia	Importancia	Posible solución
En el tablero inicial el jugador debe llevar cada bola hasta su meta. Cuando lo hace, una puerta se cierra encerrando a la bola en ella. Si el usuario hace un movimiento brusco justo en el momento en el que se está cerrando la puerta la bola se sale de la meta, pero ya se ha activado el modo “la bola ha llegado a la meta”.	En raras ocasiones	W	Hay dos posibles soluciones, bien aplicarle más velocidad a la puerta para que se cierre más rápido, o que el plano que detecta que la bola se encuentra dentro de la meta, esté en la pared que está más alejada de la entrada a la meta.

Tabla 18. ERROR 4.

ERROR 5: Encontrado en la pantalla 3.

Descripción	Frecuencia	Importancia	Posible solución
Cuando la bola se encuentra en el plano antigravedad y se sale de él, otro plano lo detecta y sube con la misma aceleración que la de la gravedad para simular la antigravedad. Para detectar la “caída” colocamos un plano justo encima del plano antigravedad, de tal manera que cuando el plano que sube colisiones con este se indica que la bola se ha “caído”. Esta detección no se realizaba.	Ocasionalmente	E	Crear un plano que se emparentará con la bola, es decir, la bola se quedará pegada a este, que será el plano que suba con la aceleración de la gravedad. Y detectar la colisión del plano que activa el estado GAME OVER, con la bola en vez de con el plano.

Tabla 19. ERROR 5.

5.5.2. Test de jugabilidad.

En este estudio partimos del mismo grupo de personas. El mundo de los videojuegos ha ido creciendo mucho y muy rápido en estos años, por eso es necesario hacer una diferenciación entre las personas que van a realizar los test, sobre todo, en función de la edad de estas. No reacciona de la misma manera una persona de 20 años que una de 40, puesto que el conocimiento de la persona más joven sobre los videojuegos es mayor. Por tanto, hemos realizado las siguientes diferenciaciones:

- Personas mayores de 30 años. Estas personas han interactuado con el mundo de los videojuegos pasada la adolescencia. La mayoría de ellos tenían otras formas de ocio, ya que la tecnología de los videojuegos no estaba al alcance de todos.
- Personas menores de 30 años. La tecnología ha formado parte de sus vidas casi desde la infancia, por lo tanto su grado de interacción con los videojuegos ha sido mayor.

Cada una de ellas completaba totalmente el juego para luego contestar a un test¹¹ compuesto por 19 preguntas, divididas en los siguientes puntos:

- **Análisis de la jugabilidad.** Las preguntas de esta sección analizan algunas de las características desarrolladas en el apartado 4.2. La persona que

¹¹ ANEXO III. TEST.

realiza el test evalúa las preguntas con un “1” en el caso en el que no le haya parecido bien ese aspecto, y con un “5” en caso contrario.

- Satisfacción.
 - Aprendizaje.
 - Efectividad.
 - Inmersión.
 - Motivación.
 - Emoción.
- **Análisis personal.** Se les realizan preguntas sobre la estética y los efectos sonoros, y se les pide que hagan una valoración general personal sobre el videojuego.

A continuación se muestran los resultados obtenidos mediante una serie de tablas y gráficas, que permiten evaluar los puntos débiles y fuertes, para así, poder obtener las posibles soluciones y sacar una conclusión general.

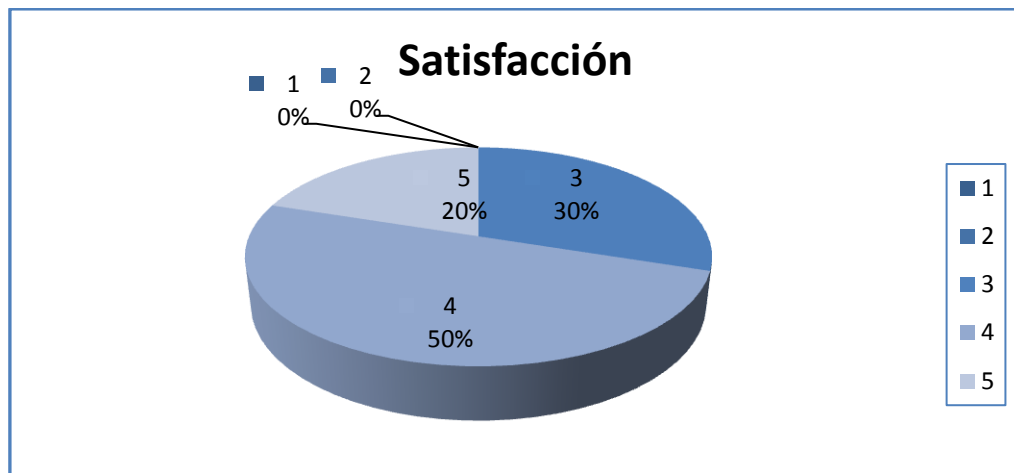
Para analizar los resultados del análisis de la jugabilidad se realiza el siguiente cálculo:

- Se calcula la media de los resultados de las preguntas que evalúan una característica para cada persona. El valor “1” representa una muy mala valoración en ese aspecto y el valor “5” una muy buena valoración.
- Se calcula la media de los resultados obtenidos en el análisis de esa característica entre todas las personas englobadas en el mismo grupo.

Por tanto, los resultados obtenidos en el análisis de la jugabilidad, para el grupo de personas que tienen una edad inferior a 30 años, son los siguientes:

Grupo: Menores de 30 años.			
Característica: SATISFACCIÓN.			
Nombre	Edad	P1	Valoración total
Álvaro	24	4	4
Clara	24	3	3
Daniel	18	5	5
Elsa	23	5	5
Jorge	24	4	4
José Luis	23	3	3
Jaime	22	4	4
Gabriel	20	4	4
Victoria	23	3	3
Rafael	19	4	4

Tabla 20. SATISFACCIÓN en menores de 30 años.

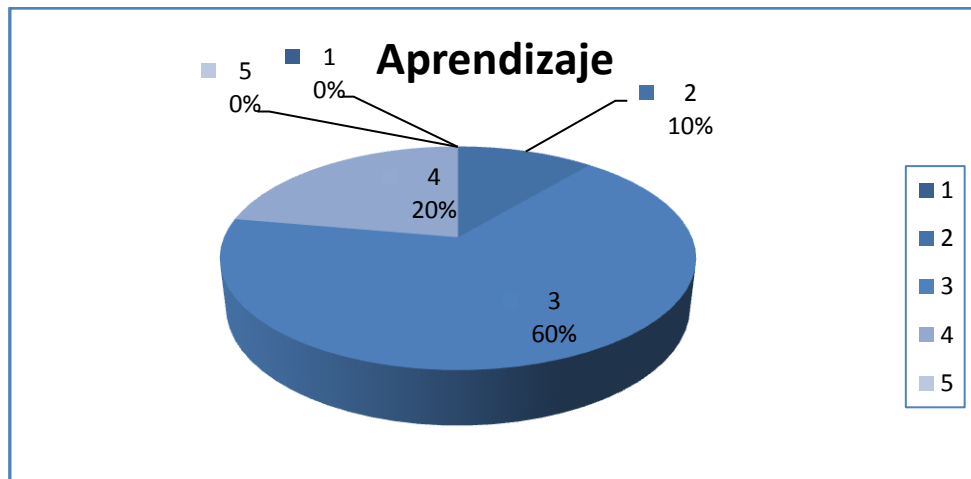


Gráfica 1. Resultados de la propiedad SATISFACCIÓN

Como se puede apreciar en la gráfica anterior, el 50% de las personas que han realizado el test ha tenido un grado de satisfacción alto mientras jugaba. Y ninguno de ellos ha valorado esta característica por debajo del “3”, con lo que podemos decir que nuestro juego atraería satisfactoriamente a personas que tengan características similares a las de este grupo. Y por tanto, la satisfacción es un punto fuerte de nuestro proyecto.

Grupo: Menores de 30 años.						
Característica: APRENDIZAJE.						
Nombre	Edad	P1	P2	P3	P4	Valoración total
Álvaro	24	5	2	3	1	3
Clara	24	5	2	2	5	3
Daniel	18	3	2	5	5	4
Elsa	23	3	1	5	1	2
Jorge	24	3	4	3	3	3
José Luis	23	5	3	1	3	3
Jaime	22	1	2	5	3	3
Gabriel	20	3	2	5	3	3
Victoria	23	4	3	4	4	4
Rafael	19	2	2	1	5	2

Tabla 21. APRENDIZAJE en menores de 30 años.



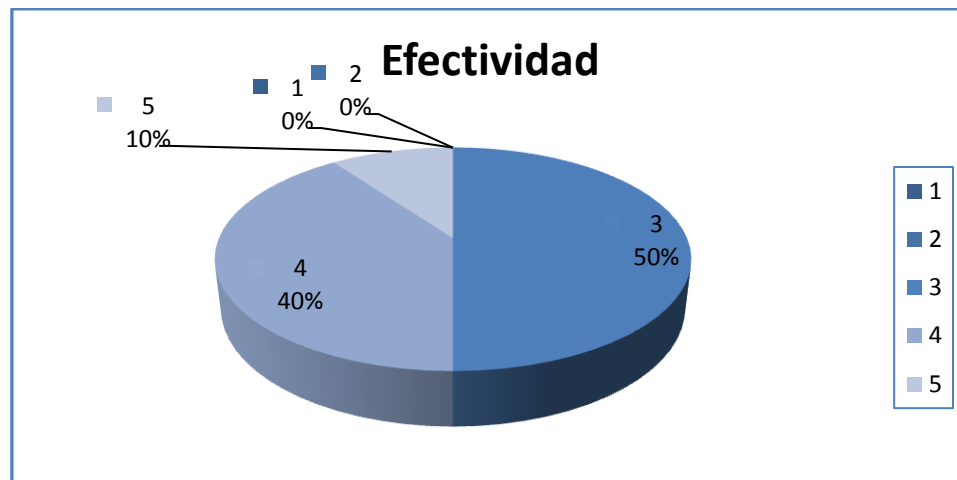
Gráfica 2. Resultados de la propiedad APRENDIZAJE.

Dentro de esta característica tenemos que valorar ciertas cosas, los valores son muchos más bajos que en el anterior análisis. El valor que predomina es el “3” con un 60%, de aquí podemos deducir que la curva de aprendizaje de nuestro juego es alta y que la mecánica no está bien descrita.

Por tanto, este es uno de los aspectos que tendremos que mejorar, ya que si la curva de aprendizaje es alta puede producir que el usuario abandone el juego antes de completarlo.

Grupo: Menores de 30 años.				
Característica: EFECTIVIDAD.				
Nombre	Edad	P1	P2	Valoración total
Álvaro	24	4	4	4
Clara	24	4	1	3
Daniel	18	5	3	4
Elsa	23	5	2	3
Jorge	24	3	5	4
José Luis	23	5	5	5
Jaime	22	5	1	3
Gabriel	20	5	1	3
Victoria	23	3	5	4
Rafael	19	3	4	3

Tabla 22. EFECTIVIDAD en menores de 30 años.

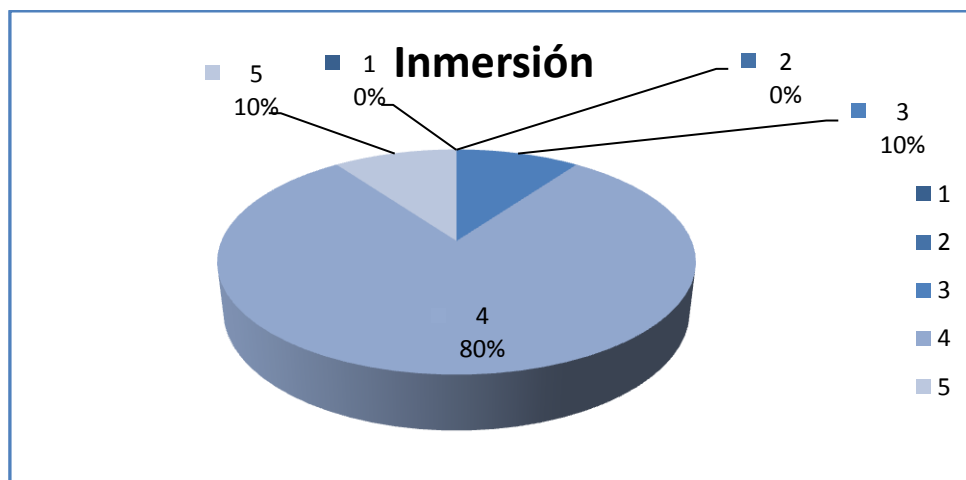


Gráfica 3. Resultados de la EFECTIVIDAD.

En cuanto a la efectividad, vemos que los resultados son buenos, un 50% ha valorado la efectividad del juego con un “3”, un 40% con un “4” y un 10% con un “5”, por lo tanto, podemos decir que el videojuego tiene un grado de captación alto y que hace que el jugador mantenga el interés por él durante la mayor parte del tiempo.

Grupo: Menores de 30 años.						
Característica: INMERSIÓN.						
Nombre	Edad	P1	P2	P3	P4	Valoración total
Álvaro	24	5	3	3	5	4
Clara	24	4	5	5	4	4
Daniel	18	5	3	4	4	4
Elsa	23	5	4	4	5	4
Jorge	24	2	2	3	5	3
José Luis	23	5	4	3	5	4
Jaime	22	5	3	3	5	4
Gabriel	20	4	5	5	5	5
Victoria	23	3	4	4	5	4
Rafael	19	4	3	5	5	4

Tabla 23. INMERSIÓN en menores de 30 años.

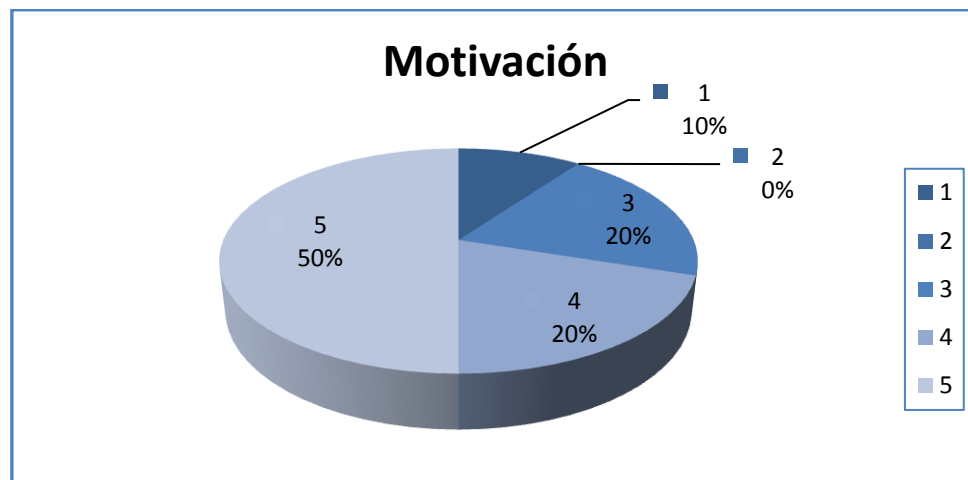


Gráfica 4. Resultado de la INMERSIÓN.

Como se aprecia en el gráfico, el 80% de las personas han empatizado fuertemente con el personaje, por lo que el grado de inmersión que produce el juego es muy alto, lo que supone un gran punto a favor para el desarrollador y por tanto, se trata de uno de los puntos fuertes de nuestro proyecto.

Grupo: Menores de 30 años.			
Característica: MOTIVACIÓN.			
Nombre	Edad	P1	Valoración total
Álvaro	24	4	4
Clara	24	3	3
Daniel	18	5	5
Elsa	23	1	1
Jorge	24	4	4
José Luis	23	5	5
Jaime	22	5	5
Gabriel	20	5	5
Victoria	23	5	5
Rafael	19	3	3

Tabla 24. MOTIVACIÓN en menores de 30 años.

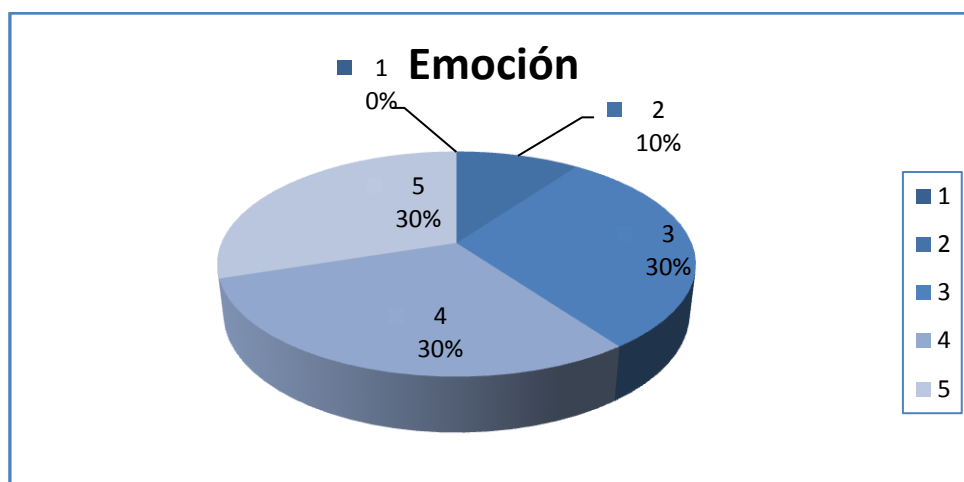


Gráfica 5. Resultados de la MOTIVACIÓN.

Los resultados en cuanto a la motivación son bastante variados, aunque predomina en un 50% el valor “5”, lo que nos indica que a la mitad de los sujetos que se han sometido al test, el videojuego les ha producido los estímulos necesarios para que su curiosidad y sus ganas de automejorar hayan aumentado en el transcurso del videojuego. Por tanto, hemos obtenidos buenos resultados en este aspecto, aunque deberíamos mejorar otros, ya que un 10% lo ha puntuado con un “1”.

Grupo: Menores de 30 años.				
Característica: Emoción.				
Nombre	Edad	P1	P2	Valoración total
Álvaro	24	3	4	3
Clara	24	5	5	5
Daniel	18	5	5	5
Elsa	23	4	5	4
Jorge	24	2	5	3
José Luis	23	1	4	2
Jaime	22	3	5	4
Gabriel	20	4	5	5
Victoria	23	3	3	3
Rafael	19	5	4	4

Tabla 25. EMOCIÓN en menores de 30 años.



Gráfica 6. Resultados de la EMOCIÓN.

Por último, los resultados obtenidos para la propiedad emoción son bastantes diferentes, aunque un 60% se ha decantado por los dos valores más altos. Esta propiedad no podemos controlarla como nos gustaría ya que tiene un grado muy alto de subjetividad.

Después del análisis de cada característica, podemos ver, en la siguiente tabla, cual de todas ellas se establece como un punto fuerte y cual un punto débil:

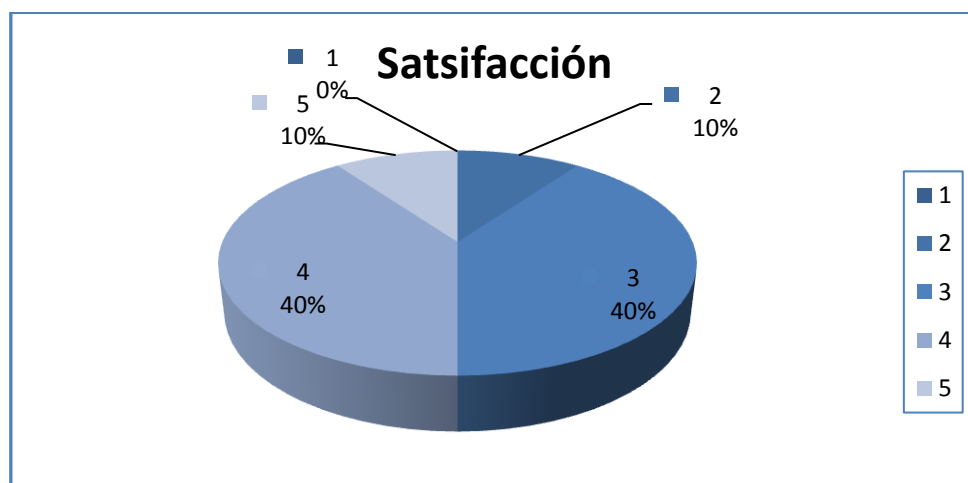
Característica	Valoración punto fuerte/débil	Mejoras
Satisfacción	FUERTE	-
Aprendizaje	DÉBIL	Poner pequeños asistentes durante el juego en función de cómo el usuario vaya superando los obstáculos, de tal manera que sirviera como un pequeño tutorial durante la primera vez que el usuario juega.
Efectividad	-	A pesar de no tratarse de un punto débil, deberíamos realizar pequeñas mejoras para aumentar la captación del jugador mediante nuevos retos y objetivos.
Inmersión	FUERTE	-
Motivación	-	Nos ocurre lo mismo que en la efectividad, sin embargo aumentar esa propiedad haría que la motivación de los jugadores para superar los niveles aumentara.
Emoción	-	Aumentar los valores de los anteriores puntos influiría notablemente en esta propiedad.

Tabla 26. Evaluación general de las características para personas menores de 30 años.

A continuación vamos a analizar los resultados para el grupo de personas mayores de 30 años y vamos a comparar los resultados con los del grupo de personas menores de 30 años:

Grupo: Mayores de 30 años.			
Característica: SATISFACCIÓN.			
Nombre	Edad	P1	Valoración total
Encarna	53	3	3
Juan Carlos	40	3	3
Rafael	53	4	4
Antonio	45	4	4
Elvira	59	5	5
Julián	42	3	3
Paloma	36	4	4
Emilio	32	4	4
Joaquín	35	3	3
José María	36	2	2

Tabla 27. SATISFACCIÓN en mayores de 30 años.

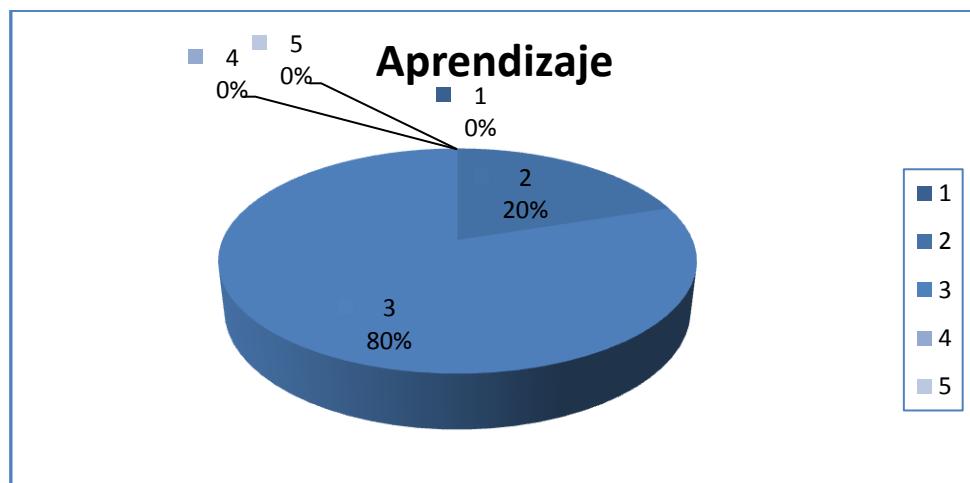


Gráfica 7. Resultados de la SATISFACCIÓN.

Como podemos apreciar en el gráfico que se encuentra arriba, los resultados de esta propiedad están bastante equilibrados, el 80 % ha valorado la satisfacción con un "3" y un "4" por lo que no son valores negativos. Sin embargo, como ya hemos visto anteriormente, las personas menores de 30 años han puntuado un poco más alta esta propiedad, con lo que podemos sacar la conclusión de que el juego les parece más divertido a las personas menores de 30 años que a las mayores de 30 años.

Grupo: Mayores de 30 años.						
Característica: APRENDIZAJE.						
Nombre	Edad	P1	P2	P3	P4	Valoración total
Encarna	53	1	2	4	4	3
Juan Carlos	40	2	2	3	3	2
Rafael	53	1	2	4	4	3
Antonio	45	1	2	4	4	3
Elvira	59	1	1	3	3	2
Julián	42	1	3	4	3	3
Paloma	36	2	2	4	4	3
Emilio	32	2	3	4	3	3
Joaquín	35	2	3	4	4	3
José María	36	2	3	4	4	3

Tabla 28. APRENDIZAJE en mayores de 30 años.

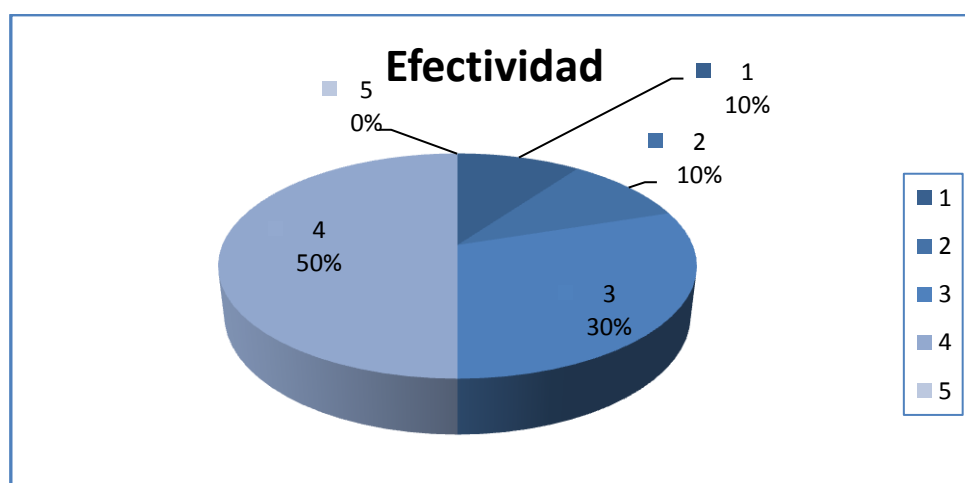


Gráfica 8. Resultados del APRENDIZAJE.

Vemos claramente que a este grupo le ha sido más difícil entender la mecánica del juego, esto se ha debido a que la mayoría de ellos no había jugado nunca a un juego similar. En contraposición, el otro grupo si conocía juegos similares y eso les ayudó a entender la mecánica y el desarrollo del juego.

Grupo: Mayores de 30 años.				
Característica: EFECTIVIDAD.				
Nombre	Edad	P1	P2	Valoración total
Encarna	53	1	2	1
Juan Carlos	40	5	3	4
Rafael	53	4	5	4
Antonio	45	2	5	3
Elvira	59	4	5	4
Julián	42	3	5	4
Paloma	36	5	2	3
Emilio	32	1	5	3
Joaquín	35	4	1	2
José María	36	4	4	4

Tabla 29. EFECTIVIDAD en mayores de 30 años.

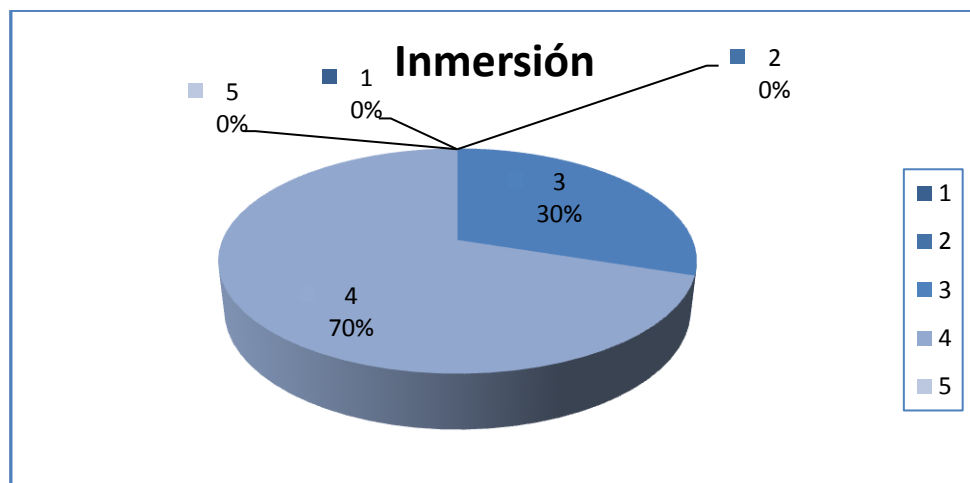


Gráfica 9. Resultados de la EFECTIVIDAD.

En esta propiedad se observa claramente que los valores son bastante más bajos que los del otro grupo, y es que el poder de captación del juego sobre estas personas es mucho menor.

Grupo: Mayores de 30 años.						
Característica: INMERSIÓN.						
Nombre	Edad	P1	P2	P3	P4	Valoración total
Encarna	53	4	3	2	2	3
Juan Carlos	40	4	2	3	5	3
Rafael	53	4	4	4	5	4
Antonio	45	3	4	4	5	4
Elvira	59	5	5	3	4	4
Julián	42	3	5	5	4	4
Paloma	36	3	3	3	4	3
Emilio	32	4	4	4	3	4
Joaquín	35	3	4	4	5	4
José María	36	5	4	4	3	4

Tabla 30. INMERSIÓN en mayores de 30 años.

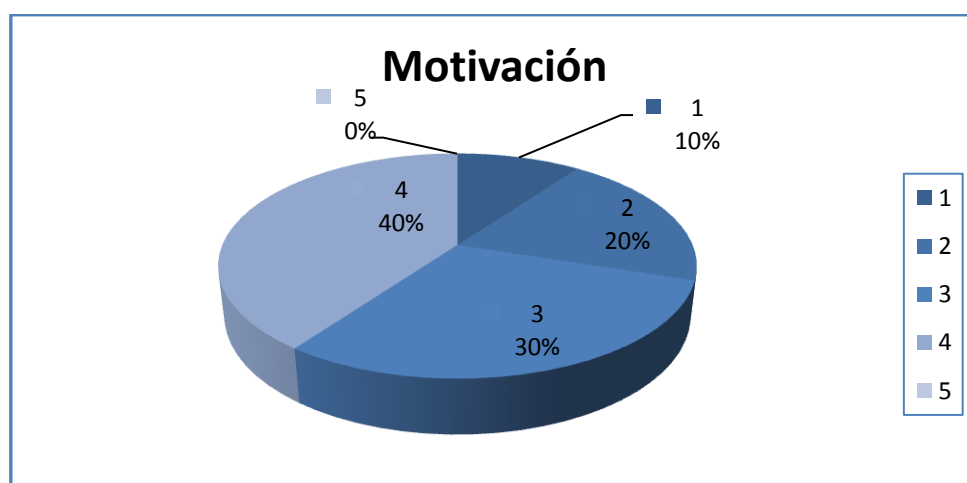


Gráfica 10. Resultados de la INMERSIÓN.

Como podemos ver en los resultados, el grado de inmersión de estas personas es mayor que en las del otro grupo, y es que, estas personas empatizan mucho más con el personaje. Esto puede deberse principalmente a que, el desconocimiento de la mecánica del juego hace que su concentración aumente.

Grupo: Mayores de 30 años.			
Característica: MOTIVACIÓN.			
Nombre	Edad	P1	Valoración total
Encarna	53	4	4
Juan Carlos	40	4	4
Rafael	53	3	3
Antonio	45	4	4
Elvira	59	3	3
Julián	42	4	4
Paloma	36	1	1
Emilio	32	2	2
Joaquín	35	2	2
José María	36	3	3

Tabla 31. MOTIVACIÓN en mayores de 30 años.

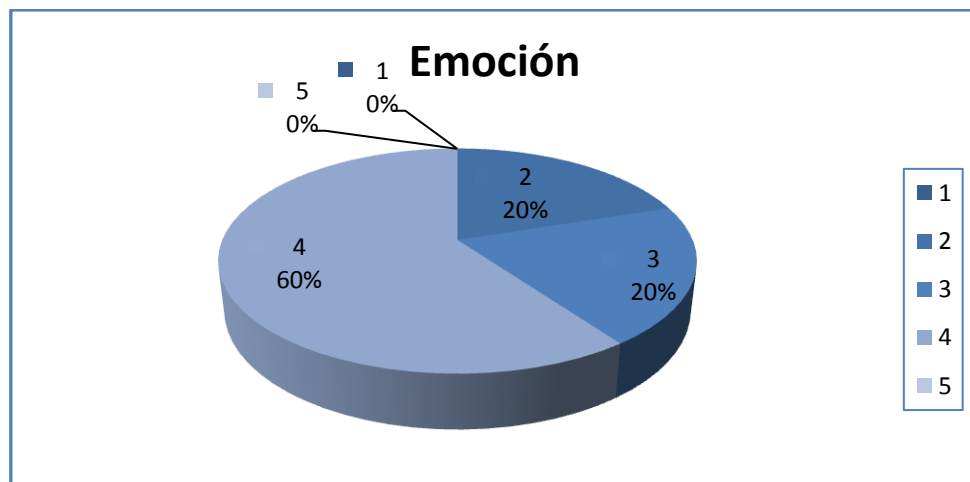


Gráfica 11. Resultados de la MOTIVACIÓN.

En cuanto a la motivación, los resultados son más bajos respecto a los del otro grupo. Estos resultados son previsibles, ya que la estimulación y la curiosidad que produce el juego en estas personas es mucho menor.

Grupo: Mayores de 30 años.				
Característica: Emoción.				
Nombre	Edad	P1	P2	Valoración total
Encarna	53	1	4	2
Juan Carlos	40	3	2	2
Rafael	53	3	5	4
Antonio	45	3	5	4
Elvira	59	4	4	4
Julián	42	5	3	4
Paloma	36	3	5	4
Emilio	32	4	5	4
Joaquín	35	2	5	3
José María	36	2	4	3

Tabla 32. EMOCIÓN en mayores de 30 años.



Gráfica 12. Resultados de la EMOCIÓN.

A pesar de los resultados anteriores, vemos que el grado de emoción que han sufrido las personas de este grupo es mayor que el del otro grupo. Esto puede ser debido a que la interacción de estas personas con juegos similares es menor y en algunos casos ninguna, con lo que el descubrimiento de nuevos aspectos hace que las emociones despertadas en el sujeto sean más intensas.

Después del análisis de los resultados en este grupo, reflejamos en la siguiente tabla los puntos fuertes y débiles:

Característica	Valoración punto fuerte/débil	Mejoras
Satisfacción	-	Debemos incluir nuevos retos, quizás relacionados con el entrenamiento de las capacidades mentales, como ejercicios de memoria.
Aprendizaje	DÉBIL	También es un punto débil en el otro grupo, por lo que las soluciones propuestas anteriormente también serían válidas. Sin embargo, deberíamos realizar un tutorial mucho más explicativo en este caso.
Efectividad	DÉBIL	Variando más lo retos quizás hacia algo mas educativo, podríamos conseguir que el grado de captación aumentase para este grupo.
Inmersión	-	La satisfacción y la efectividad van íntimamente ligadas a esta propiedad, por lo que aumentando los resultados de esas dos, probablemente conseguiremos que la inmersión sea uno de nuestros puntos fuerte.
Motivación	FUERTE	-
Emoción	FUERTE	-

Tabla 33. Evaluación general de las características para personas mayores de 30 años .

Después de este estudio, podemos sacar varias conclusiones:

- Nuestro juego es más atractivo para personas que sean menores de 30 años, luego nuestro videojuego irá enfocado a esas personas. Por tanto, las mejoras que debemos hacer deben ir enfocadas a mejorar los puntos débiles que hemos sacado durante el estudio.
- Vemos que los resultados de las propiedades motivación y emoción son mucho más altos en personas mayores de 30 años que en las menores de 30 años. El desconocimiento ante nuevos impulsos visuales y sonoros, hacen que los estímulos que reciben estas personas sean mayores y, por tanto, el grado de emoción sufrido sea superior.
- También, observamos que la propiedad aprendizaje es un punto débil en ambos estudios, y este aspecto es bastante negativo. Es muy difícil para el desarrollador la realización de un tutorial que sirva de guía para los jugadores, ya que, al haberlo diseñado, piensa que mediante los estímulos visuales y sonoros está bien explicada la mecánica del juego. A través de los

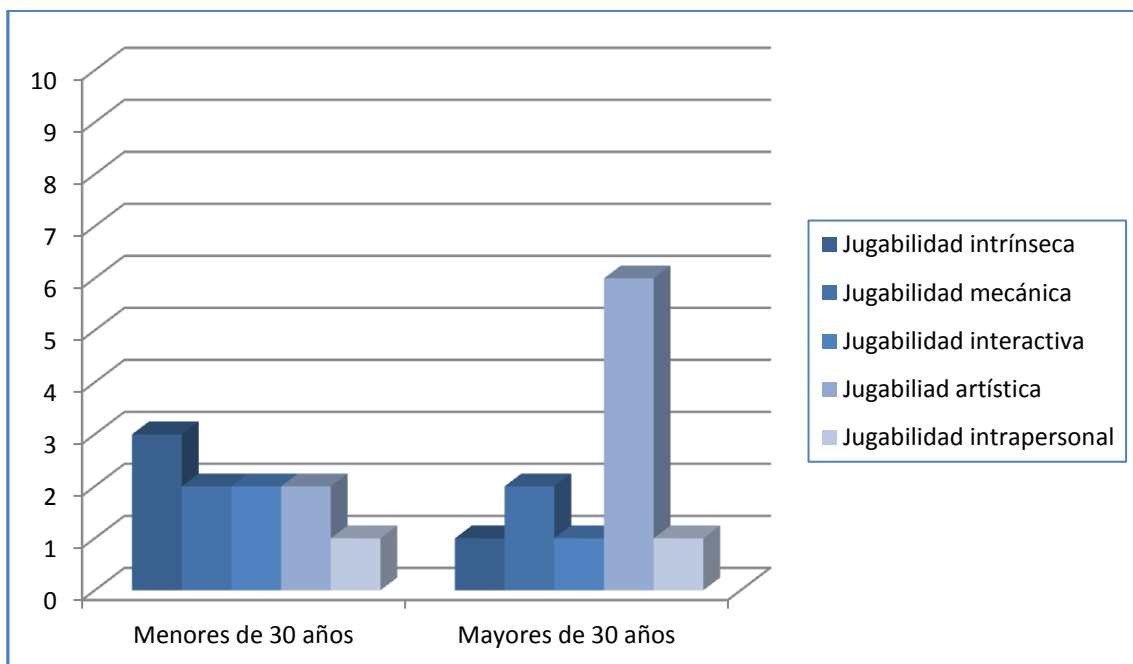
resultados, vemos, claramente, que no es así. Por tanto, habrá que dedicarle bastante tiempo a este aspecto durante la siguiente fase.

Las diferencias en los resultados entre ambos grupo se debe, en mayor parte, a que los cambios que se han producido en cuanto al ocio en ambas generaciones han sido muy drásticos, ya que en el transcurso de 20 años la tecnología del videojuego se ha desarrollado de una forma muy potente y muy rápida.

A continuación, vamos a analizar los resultados sobre el análisis personal, viendo cuales han sido las respuestas más frecuentes a las preguntas sobre qué ha sido lo que más y menos les ha gusta, las mejoras que han propuesto y el resultado final sobre la valoración general del juego, realizando la media de todas la puntuaciones.

En este análisis también vamos a analizar los resultados partiendo de la división entre personas menores de 30 años y personas mayores de 30 años.

En la siguiente gráfica se reflejan los resultados sobre lo que más les ha gustado del videojuego, cada resultado lo hemos evaluado y lo hemos caracterizado según las facetas de la jugabilidad [apartado 4.4]. El eje Y representa el número de personas que se ha sometido a la fase de testing, y el eje X los dos grupos de personas que hemos hecho.



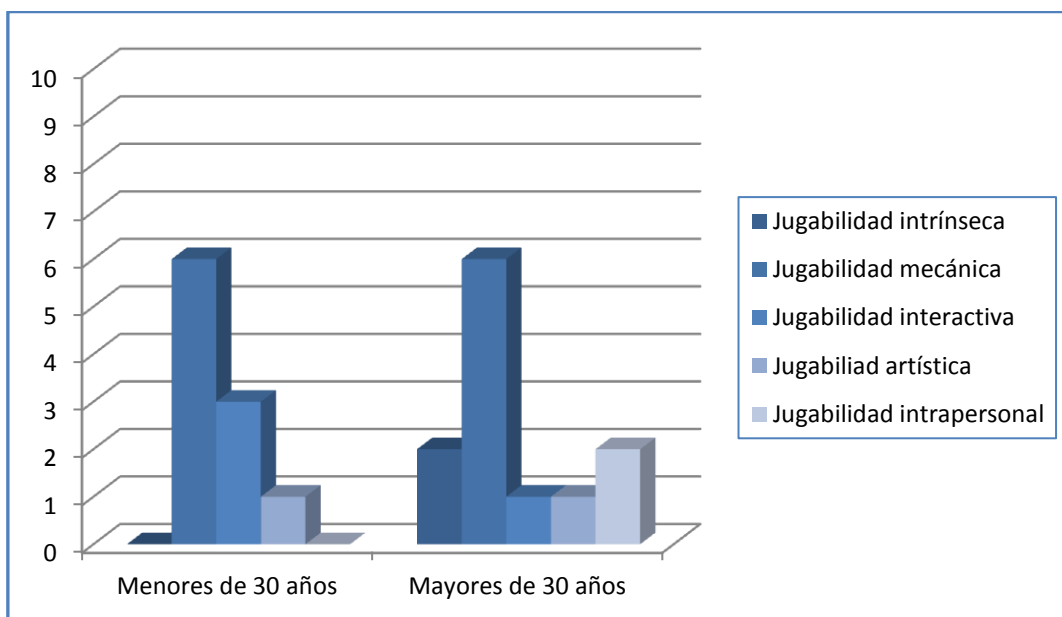
Gráfica 13. Aspectos de la jugabilidad más valorados.

Como observamos en la gráfica:

- Del grupo de menores de 30 años, han valorado positivamente las facetas de la jugabilidad el siguiente número de personas:
 - Jugabilidad intrínseca, 3 de los 10.
 - Jugabilidad mecánica, 2 de los 10.
 - Jugabilidad interactiva, 2 de los 10.
 - Jugabilidad artística, 2 de los 10.
 - Jugabilidad intrapersonal, 1 de los 10.
- Del grupo de mayores de 30 años:
 - Jugabilidad intrínseca, 1 de los 10.
 - Jugabilidad mecánica, 2 de los 10.
 - Jugabilidad interactiva, 1 de los 10.
 - Jugabilidad artística, 6 de los 10.
 - Jugabilidad intrapersonal, 1 de los 10.

Por tanto, a la mayoría del grupo de personas menores de 30 años les han gustado más aspectos como la mecánica del juego, los efectos sonoros o el comportamiento del personaje en los diferentes momentos del juego; mientras que, a los mayores de 30 años, les han gustado aspectos como la calidad gráfica o la historia del videojuego.

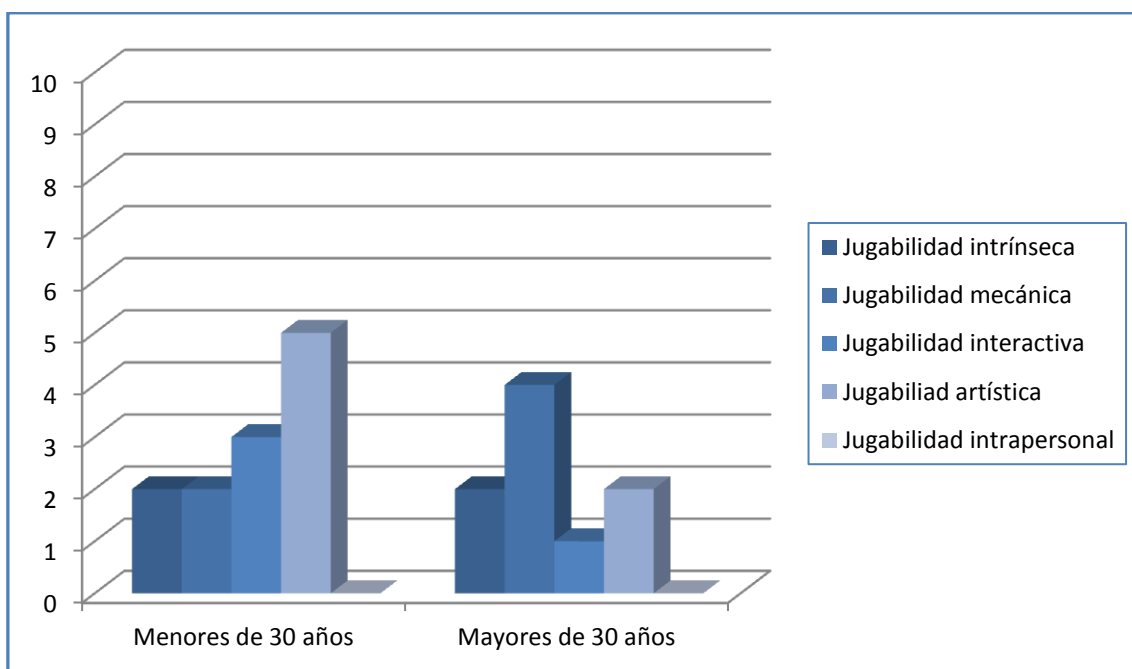
En la siguiente gráfica se recogen los resultados sobre los aspectos que menos les han gustado, englobándolos de igual manera al anterior estudio.



Gráfica 14. Aspectos de la jugabilidad menos valorados.

Como se puede apreciar, la mayoría de los aspectos negativos están centrados en la jugabilidad intrínseca. Esto se debe, como ya hemos visto en el análisis de la jugabilidad, a los problemas que han tenido la mayoría para entender la mecánica del juego.

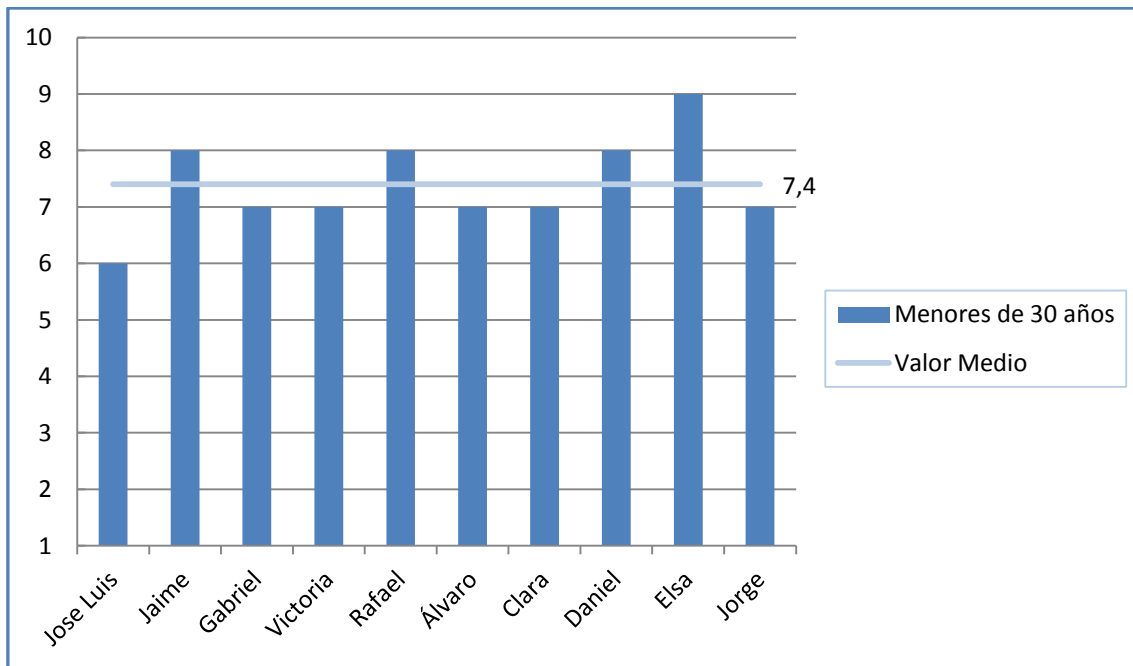
En el gráfico que tenemos a continuación vamos a ver qué tipo de jugabilidad es en la que nos han propuesto mejoras. Como podemos apreciar, los resultados son dispares y es que, las mejoras aportadas han sido muchas.



Gráfica 15. Aspectos de la jugabilidad a mejorar.

Se pueden ver todos los test realizados en el ANEXO.

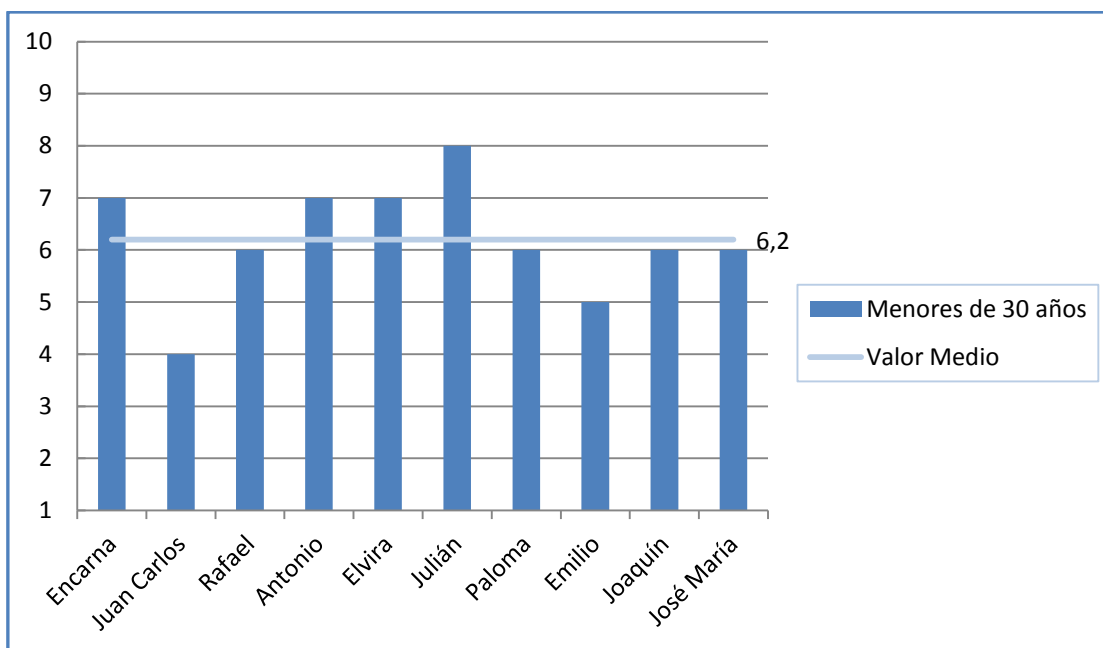
Y por último, vamos a evaluar la valoración general obtenida de nuestro videojuego tras la realización de la fase de testing.



Gráfica 16. Valoración general del videojuego.

El valor medio obtenido como resultado de las puntuaciones del estudio a las personas que forman el grupo de menores de 30 años ha sido de 7.4, lo que es una valoración bastante alta y por lo que podemos estar orgullosos del resultado de la primera fase, pero debemos seguir trabajando para mejorar los aspectos que hacen que esta valoración no sea de 10.

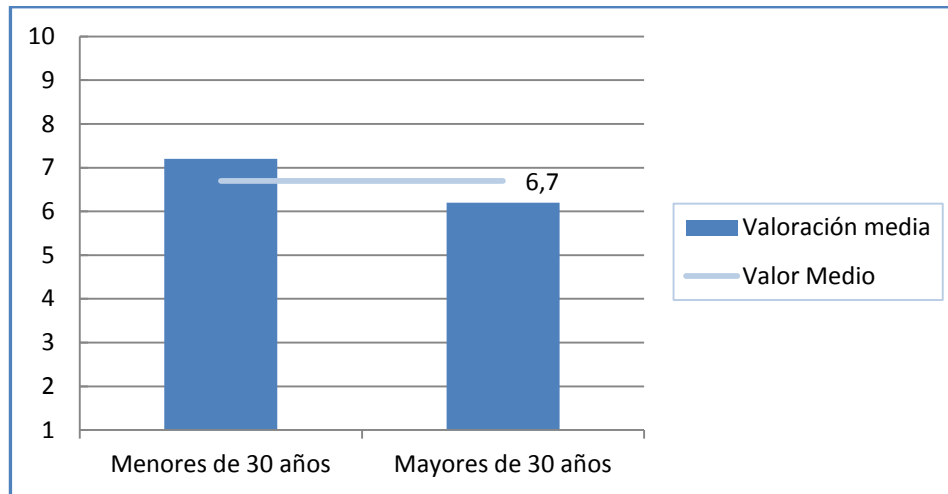
El siguiente gráfico muestra las valoraciones y el valor medio de las personas mayores de 30 años.



Gráfica 17. Valoración general del videojuego.

El valor medio de las puntuaciones de las personas del grupo mayores de 30 años es de 6.2.

Luego la media total y por tanto la valoración total del videojuego, se representa en la siguiente gráfica.



Gráfica 18. Valoración total.

5.6. CONCLUSIÓN DE LA FASE DE TESTING

Como conclusión de la fase de testing, podemos determinar que con este estudio hemos conseguido:

- Encontrar el público potencial de nuestro videojuego, que como hemos visto, mediante los resultados y las gráficas, se trata de personas menores de 30 años y que tengan un perfil similar a los de las personas que hayan valorado más positivamente el videojuego.
- Determinar cuáles son los puntos fuertes y débiles del juego, procedería a realizar los cambios oportunos con mantener los puntos fuertes y conseguir que los puntos débiles sean considerados fuertes en el futuro.
- Detectar pequeños errores durante la ejecución del videojuego que debemos corregir para que el producto final sea impecable.

6. CONCLUSIONES Y LÍNEAS FUTURAS

A continuación, se va a exponer una conclusión personal, para determinar los conceptos aprendidos y hacer una valoración personal del trabajo realizado. Además, se explicará la planificación de las fases del proyecto, el tiempo total invertido en la realización de éste mediante el diagrama de GANTT y las fases en las que se ha dividido el proyecto con el diagrama de PERT.

6.1. SOBRE LA REALIZACIÓN DEL PROYECTO.

La realización de este proyecto ha significado la visión de una nueva rama de la ingeniería totalmente diferente a la vista anteriormente, donde a pesar de tener que invertir un gran número de horas, se trata de un trabajo que satisface el hambre por hacer algo grande y la capacidad de hacerlo como y cuando quieras.

El conocimiento de una herramienta como Blender en el momento en el que nos encontramos, donde la animación 3D nos bombardea haya donde vayamos, es una experiencia excitante y que te hace valorar el trabajo que hay detrás de aquello que ves visualmente sin esfuerzo.

Por otro lado, integrarse en el mundo de los videojuegos, conocer su historia, su evolución y poder contribuir, muy humildemente, a este gigante, ha sido una de las experiencias más enriquecedoras de mi vida académica. Además, realizar un proyecto tan ambicioso como un videojuego, me ha hecho entender la cantidad de conocimientos no sólo técnicos, sino sociológicos que hacen que un videojuego se convierta en un fenómeno social. Y también, poder enfocar la escritura de mi proyecto hacia este aspecto, acercando a los futuros estudiantes una parte, que a pesar de ser la más importante dentro de este mercado, es una de las que más pasa inadvertida.

6.2. PLANIFICACIÓN DEL PROYECTO

La elección de la temática de un proyecto es la parte más complicada, la oferta que se ofrece en la universidad es muy amplia y es necesario realizar una búsqueda para encontrar el que más motive. Una vez encontrado, la planificación del trabajo en un proyecto como este es muy importante. Es necesario dividir el proyecto en diversas partes para obtener un resultado satisfactorio.

6.2.1 Búsqueda de las posibles tecnologías.

Hay que hacer un estudio de las diversas tecnologías que hay en el mercado para la realización del proyecto, en este caso en concreto el desarrollo de un videojuego. Al ser una industria relativamente novedosa, la mayoría de la documentación se encuentra en la red. En el caso de este proyecto, ya se conocía desde un principio la tecnología que se iba a utilizar, sin embargo, era necesario hacer el estudio, puesto que se tendrían que recoger en esta memoria. El hecho de hacerlo antes del desarrollo técnico del videojuego, supuso entrar en este mundo desde el punto de vista del desarrollador y tener en cuenta aspectos, como algunos de los que se recogen en el apartado de diseño (5.3), que de otra manera hubieran pasado inadvertidos.

6.2.2 Iniciación con la herramienta Blender.

Los comienzos con Blender son bastante duros. La curva de aprendizaje de la aplicación es muy acentuada al principio, para suavizarse transcurridos unos cuatro meses. El usuario se enfrenta a una aplicación con una interfaz gráfica cargada de ventanas y éstas a su vez, con infinidad de botones. El manejo correcto de todas las opciones que Blender te proporciona, permite que el trabajo realizado con él tenga una calidad claramente notable.

Durante los primeros meses, estás continuamente cometiendo los mismos errores. La mejor manera de comenzar es realizando pequeños modelos, texturizándolos e implementándolos. Primeros modelos que han supuesto un gran esfuerzo y que tienen un resultado dudoso. Se aprecia claramente que la diferencia entre el primer nivel y el último es bastante notable. No sólo porque la finalidad del juego sea ir aumentando la dificultad con la que se enfrentará el jugador, sino también, porque el nivel de conocimiento a lo largo del desarrollo incrementó rápidamente. Cada vez que se modela algo nuevo, es necesario el estudio de mucha documentación y realizarlo una y otra vez, hasta dominar la técnica de modelado que se está utilizando. Más tarde, se observa que el uso de esa técnica estudiada no entrañará gran dificultad.

6.2.3 Desarrollo del modelado.

Una vez que se está familiarizado con la aplicación y que se han realizado pequeños modelos de iniciación que se encuentran en tutoriales, se procede a examinar los bocetas realizados previamente a papel para iniciar el desarrollo del modelo en 3D. Cuando el modelo está terminado, se continúa con la texturización, se procede a la integración de las cámaras y las luces que se.

6.2.4 Desarrollo de la implementación.

Terminado el desarrollo del modelo 3D, se puede comenzar con la implementación. Esta fase es un poco más complicada que la anterior porque personalmente todas las implementaciones que se habían realizado anteriormente habían sido programáticamente, y en *Blender*, se hace de forma visual. Aunque, a priori parezca más sencillo, no lo es, ya que es necesario diferenciar la implementación de cada objeto por separado y dividir la implementación en ciclos muy diferenciados que se realimentan a través de mensajes.

6.2.5 Perfeccionamiento.

Una vez se terminó el desarrollo de los tres niveles, la soltura en el manejo de Blender es notable, por lo que, se procede a la mejora de aspectos en los escenarios que la primera vez que se realizaron no tuvieron el resultado esperado.

6.2.6 Fase de testing.

La fase de testing es, quizás, la fase más importante de todas. Primeramente, le da la oportunidad al desarrollador a percatarse de errores que de otra manera hubiera pasado por alto. Y por otro lado, realiza un estudio bastante amplio de la parte social y psicológica del videojuego, estudiando las reacciones que éste despierta en un grupo de sujetos, y sacando conclusiones que permiten mejorar el proyecto.

6.2.7 Redacción de la memoria.

Por último, la redacción de la memoria. Al ser estudiante de ciencias, ha resultado realmente complicada la escritura, puesto que, desgraciadamente, durante la carrera no se realizan ejercicios de redacción. Durante esta fase, se ha

observado lo difícil que es expresar con claridad lo que se quiere exponer, en qué cuantía y con qué detalle.

6.3. ESTIMACIÓN FINAL.

Una vez desarrollado todo el proyecto, podemos calcular la estimación real. Como vemos en la siguiente tabla, las tareas que habíamos dividido en la estimación inicial han sufrido cambios. Estos cambios representan un desglose de cada una de las tareas principales, esto se debe a que a la hora de realizar el proyecto, nos hemos percatado de que cada tarea inicial ha tenido una complejidad muy grande y que ha sido necesario su desglose para el buen desarrollo del proyecto.

Todo esto se ve reflejado en la siguiente tabla, donde con cada nueva tarea viene asociado su tiempo estimado con el 100% de la tarea realizada, es decir, el tiempo necesario para la realización de esa tarea.

Como se puede observar el tiempo total necesario es muy superior al estimado inicialmente, tiempo calculado en el apartado 4.6., esto se debe, a las nuevas tareas que han ido surgiendo a lo largo del desarrollo. El aprendizaje de *Blender* es una de ellas, ya que, hasta que no tienes que enfrentarte a la herramienta no puedes realizar una buena estimación.

Tarea	Fecha de inicio	Fecha de fin	Tiempo estimado (días laborales)
Documentación Blender	2/03/10	6/04/10	25
Documentación para la memoria	14/03/10	2/04/10	15
Escritura de la memoria	4/04/10	28/05/10	40
Aprendizaje Blender	6/04/10	20/07/10	75
Concepción histórica	3/05/10	21/05/10	14
Desarrollo del diseño en papel	21/05/10	18/06/10	20
Modelado 3D: Plataforma 1	20/07/10	31/08/10	30
Modelado 3D: Plataforma 2	31/08/10	5/10/10	25
Modelado 3D: Plataforma 3	5/10/10	2/11/10	20
Texturizado: Plataforma 1	2/11/10	4/11/10	2
Texturizado: Plataforma 2	4/11/10	9/11/10	3
Texturizado: Plataforma 3	9/11/10	12/11/10	3
Implementación: Plataforma 1	12/11/10	10/12/10	20
Implementación: Plataforma 2	10/12/10	21/01/11	30
Implementación: Plataforma 3	21/01/11	25/02/11	25
Integración sonidos	25/02/11	28/02/11	1
Modelado 3D: Menús	28/02/11	4/03/11	4
Implementación: Menús	4/03/11	11/03/11	5
Diseño del test de jugabilidad	11/03/11	14/03/11	1
Fase de testing	14/03/11	23/04/11	30
Resultados fase testing	25/04/11	4/05/11	7

Escritura de las conclusiones de la fase de testing	30/05/11	4/06/11	5
TOTAL:			400

Tabla 34. Estimación final.

A continuación vemos el diagrama de GANTT y de PERT¹² asociado a la estimación final.

¹² Ver ANEXO V.

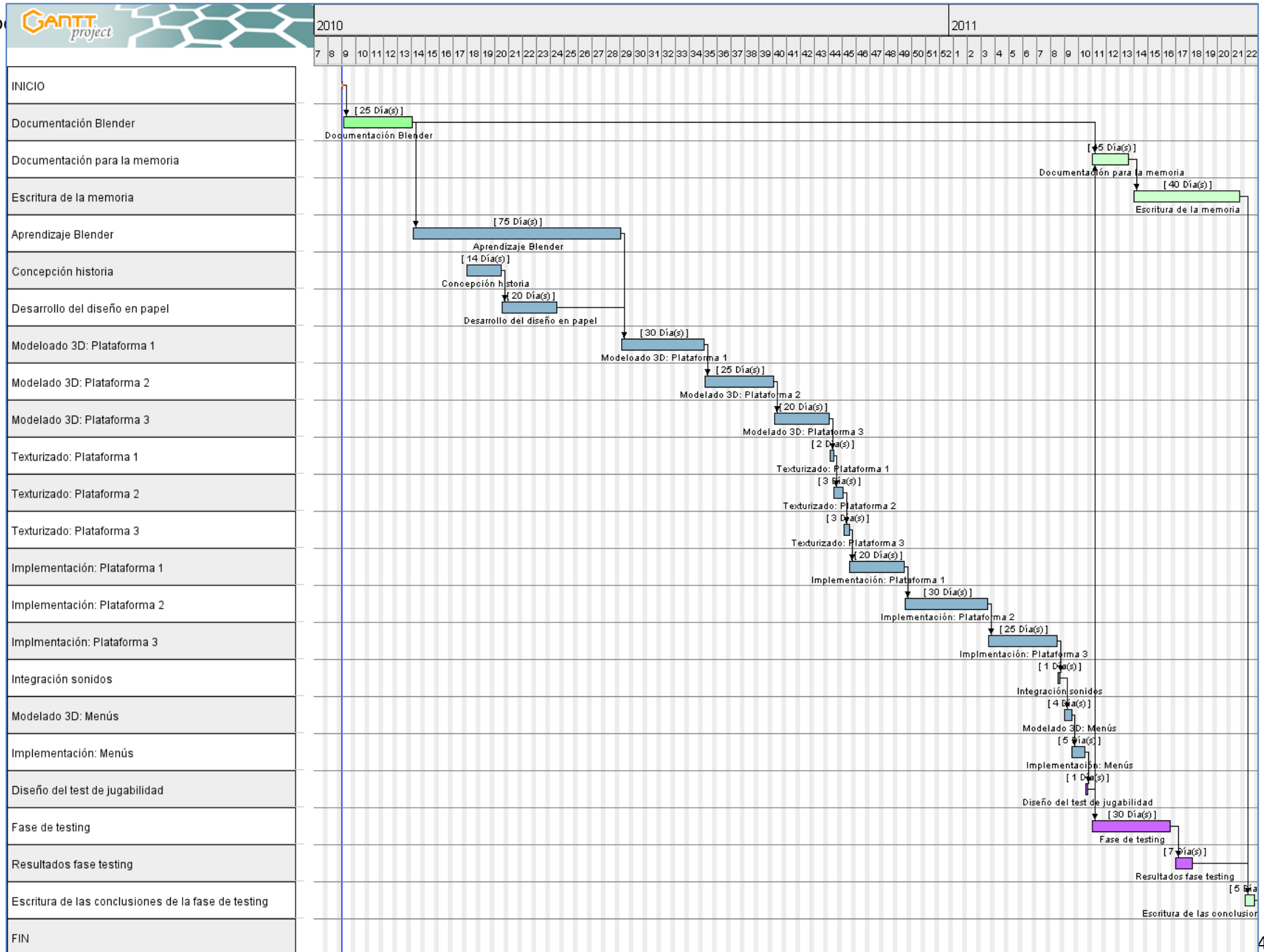



Tabla 35. Diagrama de GANTT final.

6.4. PRESUPUESTO FINAL.

Habiendo calculado en el apartado la estimación real, se puede realizar el presupuesto final. Puesto que el presupuesto es proporcional a la estimación, es normal que el cálculo del total haya aumentado.

		UNIVERSIDAD CARLOS III DE MADRID Escuela Politécnica Superior			
PRESUPUESTO FINAL DE PROYECTO					
1.- Autor: Ana Román Rodríguez					
2.- Departamento: Informática					
3.- Descripción del Proyecto: Implementación de un videojuego con la herramienta Blender					
- Título					
- Duración (meses): 2					
Tasa de costes Indirectos: 20%					
4.- Presupuesto total del Proyecto (valores en Euros): 26.000€					
5.- Desglose presupuestario (costes directos):					
PERSONAL					
Apellidos y nombre	Categoría	Dedicación (hombres mes) ^{a)}	Coste hombre mes	Coste (Euro)	Firma de conformidad
				0,00	
Peralta Donante Juan	Ingeniero Sénior	3.05	4.289,54	13.083,10	
Román Rodríguez Ana	Ingeniero	3.05	2.694,39	8.217,89	
				0,00	
				0,00	
	Hombres mes	6,1	Total	21.300,99	
^{a)} 1 Hombre mes = 131,25 horas. Máximo anual de dedicación de 12 hombres mes (1575 horas) Máximo anual para PDI de la Universidad Carlos III de Madrid de 8,8 hombres mes (1.155 horas)					

EQUIPOS					
Descripción	Coste (Euro)	% Uso dedicado proyecto	Dedicación (meses)	Periodo de depreciación	Coste imputable ^{d)}
		100		60	0,00
		100		60	0,00
		100		60	0,00
		100		60	0,00
		100		60	0,00
					0,00
					Total: 0,00
^{d)} Fórmula de cálculo de la Amortización: <div style="display: flex; align-items: center;"> <div style="margin-right: 20px;"> $\frac{A}{B} \times C \times D$ </div> <div> <p>A = nº de meses desde la fecha de facturación en que el equipo es utilizado</p> <p>B = periodo de depreciación (60 meses)</p> <p>C = coste del equipo (sin IVA)</p> <p>D = % del uso que se dedica al proyecto (habitualmente 100%)</p> </div> </div>					
SUBCONTRATACIÓN DE TAREAS					
Descripción	Empresa	Coste imputable			
					Total: 0,00
OTROS COSTES DIRECTOS DEL PROYECTO ^{e)}					
Descripción	Empresa	Costes imputable			
					Total: 0,00
^{e)} Este capítulo de gastos incluye todos los gastos no contemplados en los conceptos anteriores, por ejemplo: fungible, viajes y dietas, otros,...					
6.- Resumen de costes:					
Presupuesto Costes Totales		Presupuesto Costes Totales			
Personal		21.301			
Amortización		0			

Subcontratación de tareas	0
Costes de funcionamiento	0
Costes Indirectos	4.260
Total	25.561

6.5. LÍNEAS FUTURAS

El desarrollo de un proyecto fin de carrera de este tipo nos ofrece la oportunidad de enfocarlo hacia proyectos futuros más ambiciosos.

La primera posibilidad es continuar mejorando el videojuego, de tal manera que esta pequeña versión beta pueda evolucionar hacia algo más profesional. Para ello, primeramente, deberíamos mejorar los gráficos y el texturizado, es decir mejorar la apariencia visual del juego. Más tarde, aumentar la sensibilidad de los controles y cambiar los efectos de sonido actuales por otros más creíbles y de mayor calidad. También deberíamos realizar una interfaz gráfica más dinámica y, por supuesto, un sistema de puntuación donde el jugador pueda ir guardando los niveles superados con sus mejores marcas.

En el caso de querer continuar con el proyecto, para intentar comercializarlo, deberíamos diseñar nuevos niveles con una mayor complejidad de formas y de obstáculos. También, podríamos integrar enemigos móviles, aumentando la dificultad del juego. O incluso, añadirle a nuestro personaje nuevas habilidades, como podría ser incrementar la velocidad en determinados momentos o ser inmune en un determinado espacio de tiempo a los enemigos que se encuentre en su camino.

7. BIBLIOGRAFÍA

Libros utilizados

- Tony Mullen, Ton Roosendaal y Bassam Kurdali. Introducing Character Animation with Blender.
- Jeff Ward (2008). "What is a Game Engine?" Game Career Guide.
- González Sánchez, J. L. (2010) Tesis doctoral: "Jugabilidad. Caracterización de la experiencia del Jugador en Videojuegos". Universidad de Granada.

Motores de juego

- <http://www.naturalmotion.com/euphoria.htm>.
- http://www.maximumpc.com/article/features/3d_game_engines?page=0,0
- Paul Lilly (2009). "Doom to Dunia. A Visual History of 3D Game Engines". www.maximumpc.com
- http://www.maximumpc.com/article/features/3d_game_engines?page=0,1
- http://www.gamecareerguide.com/features/529/what_is_a_game_engine.php
- http://www.worldlingo.com/ma/enwiki/en/Game_engine
- <http://www.eslultima.com/breve-historia-de-motores-de-juegos-episodio-ii/>
- <http://www.devmaster.net/engines/list.php>
- <http://www.develop-online.net/news/32250/The-top-10-game-engines-revealed>

Modelado 3D

- <http://www.slideshare.net/gbgarcia/herramientas-de-modelado-3d-2455690>
- <http://www.slideshare.net/guestd52308/modelado-de-personajes-3d>
- http://www.3dm.am/en/history_of_3d_modeling.html
- <http://www.dgcg.co.uk/History.html>
- http://www.ehow.com/facts_5070359_history-computer-animation.html
- http://hem.passagen.se/des/hocg/hocg_1960.htm
- http://wiki.cgsociety.org/index.php/Comparison_of_3d_tools
- http://www.tdt3d.be/articles_viewer.php?art_id=99

Blender

- <http://www.Blender.org/>
- <http://www.Blender.org/features-gallery/features/>
- <http://www.Blender.org/development/>

- <http://www.Blender.org/community/user-community/>
- <http://www.yofrankie.org/>
- <http://fisicomolon.blogspot.com/2009/06/curso-creacion-de-videojuegos-en.html>
- <http://blender.gulo.org/>
- [http://es.wikibooks.org/wiki/Blender_3D: novato a profesional](http://es.wikibooks.org/wiki/Blender_3D:_novato_a_profesional)
- <http://apuntesdeblender.iespana.es/>

8. REFERENCIAS

- [1]. <http://www.tron-sector.com/articles/view.aspx?id=30>
- [2] <http://www.britannica.com/EBchecked/topic/163303/Digital-Equipment-Corporation-DEC>
- [3] <http://www.ilm.com/>
- [4] <http://www.d2.com/>
- [5] <http://www.rhythm.com/home/>
- [6] <http://www.pixar.com/>
- [7] <http://www.autodesk.com/maya>
- [8] www.softimage.com
- [9] www.autodesk.com/3dsmax
- [10] www.newtek.com
- [11] www.blender.org
- [12] www.maxon.net
- [13] www.sidefx.com
- [14] www.rhino3d.com
- [15] www.cheta3d.com
- [16] www.pixologic.com/home.php
- [17] <http://www.nvidia.es/page/home.html>
- [18] <http://www.amd.com/es/Pages/AMDHomePage.aspx>
- [19] <http://www.microsoft.com/es/es/default.aspx>
- [20] <http://www.epicgames.com/>
- [21] <http://www.idsoftware.com/>
- [22] <http://www.opengl.org/>
- [23] <http://www.lith.com/>
- [24] <http://www.directx.es/>
- [25] <http://blog.criteriongames.com/>
- [26] <http://www.crytek.com/>
- [27] <http://www.rockstargames.com/>
- [28] <http://www.naturalmotion.com/euphoria>
- [29] <http://www.ogre3d.org/>
- [30] <http://www.genesis3d.com/>
- [31] <http://www.garagegames.com/products/torque-3d>
- [32] <http://source.valvesoftware.com/>

ANEXOS

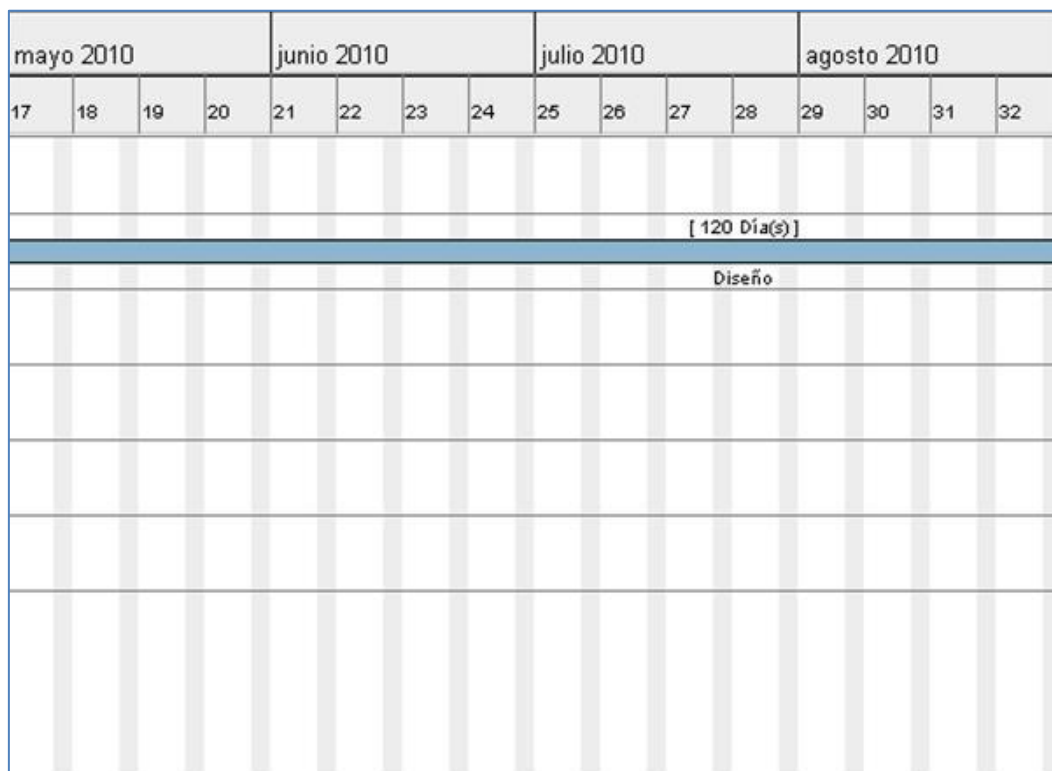
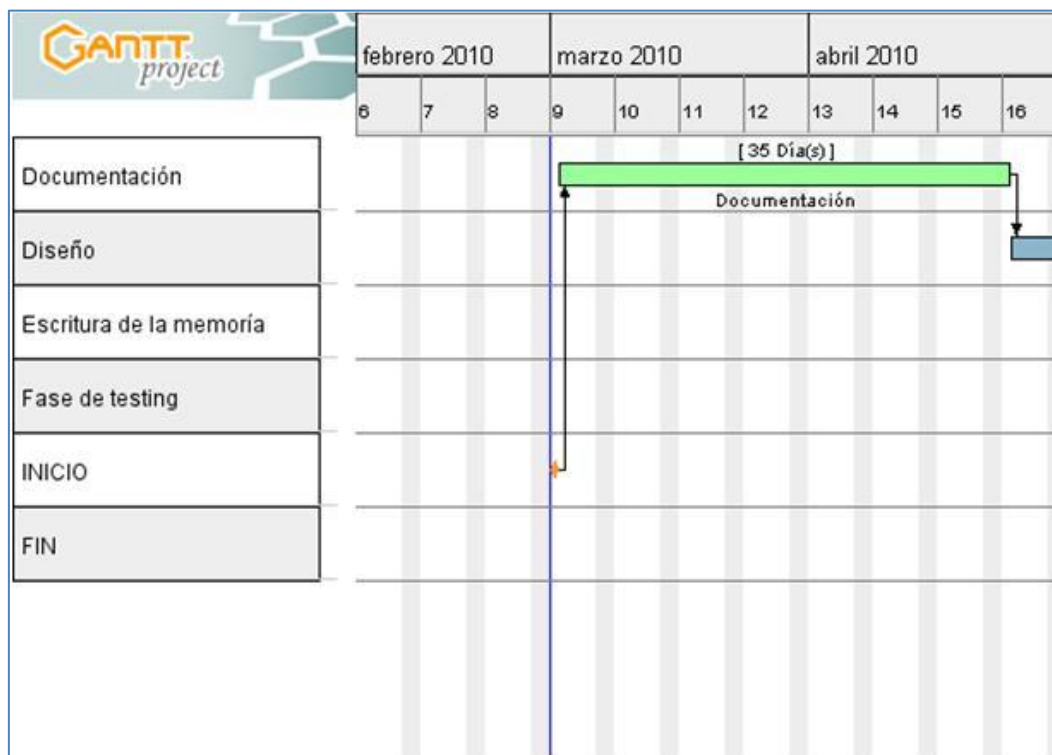
ANEXO I. Acrónimos y definiciones.

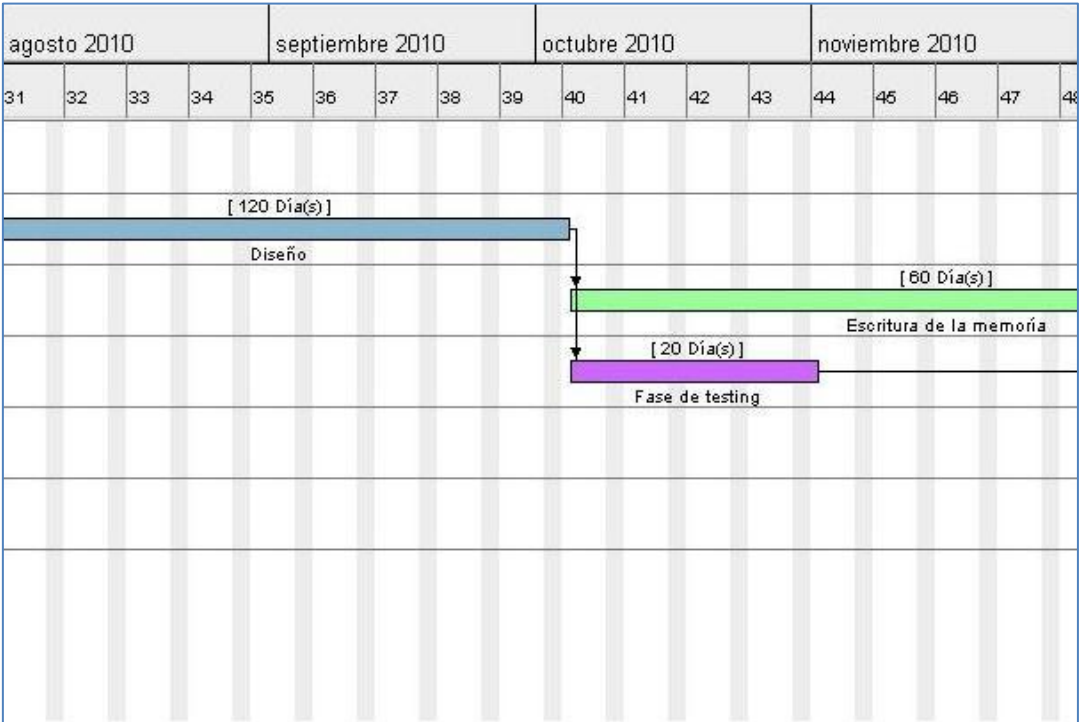
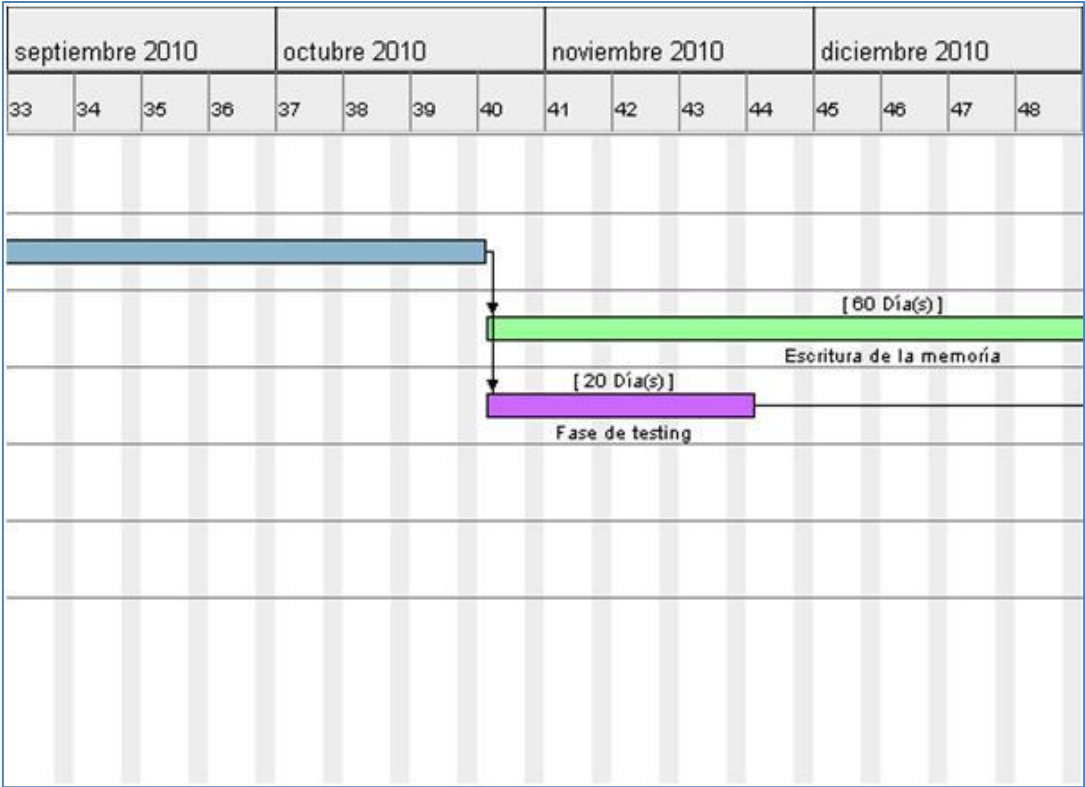
- **Sprite.** Se aplica este término a imágenes que son tanto bidimensionales como tridimensionales y que formando parte de una secuencia, constituyen una animación incluida dentro de las escenas de un videojuego.
- **Renderizado.** Proceso que realiza un ordenador para generar una imagen bidimensional a partir de un modelo generado en 3D.
- **Rag-doll physics.** Es un procedimiento usado en animación para remplazar animaciones que están consideradas como “cuerpos estáticos muertos”. Es decir, controlar el movimiento del esqueleto de los personajes que salen en los juegos, para que no parezcan “muñecos de trapo” y que las acciones que hagan durante el juego sean lo más realistas posible.
- **Shader.** Son instrucciones software utilizadas para el renderizado. Se trata de cálculos que proporcionan al programador el poder realizar la representación gráfica de luces, sombras o partículas, muy útiles a la hora de crear fuego o niebla.
- **Bump mapping.** Técnica que se aplica a cada pixel, por la que se produce una perturbación en la superficie de los objetos que están siendo renderizados. De esta manera, hace que la representación de las superficies de los objetos sea mucho más detallada.
- **Normal mapping.** Técnica que se encarga de poner una textura especial en los objetos de tal manera que genera un desplazamiento perpendicular en los polígonos que lo componen.
- **Specular highlight.** Es una textura especial que define la cantidad de brillo de una superficie.
- **API** (Interfaz de Programación de Aplicaciones). Grupo de rutinas que proporciona un sistema operativo, una aplicación o una librería, que definen como invocar un servicio desde un programa que estos nos dan. Es decir, no es más que una interfaz de comunicación entre componentes software.

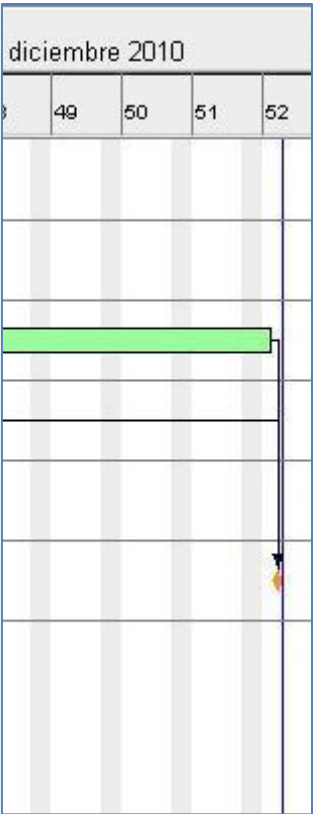
- **SDK** (Kit de Desarrollo Software). Conjunto de bibliotecas, APIs y herramientas para la programación en sistemas operativos y servicios.
- **FPS** (first-person shooters). Genero de juego nacido en los años 70 donde el jugador observa el mundo desde una perspectiva en primera persona, observando la escena desde los ojos del protagonista.
- **drag-and-drop**. Técnica “arrastrar y soltar” para la creación de espacios gráficos. Esto facilita el desarrollo quitando complejidad de implementación al usuario.
- **Game Mechanics**. La mecánica del juego define las operaciones que se pueden realizar en el videojuego, fijándose también los objetivos a los que el jugador debe llegar, las recompensas al conseguirlos o las condiciones que se deben dar para obtener la victoria.
- **Python**. Lenguaje de programación de alto nivel, orientado a objetos y con tipado dinámico.
- **Sandbox**. Se trata de una técnica informática por la cual se aíslan procesos con el fin de ejecutar diversos programas de manera separa y eficiente.
- **Unwrapping**. Técnica por la cual se coloca una imagen en un objeto modelado o en una o varias caras de este, para que gráficamente veamos el objeto como si estuviera empapelado con ella.
- **Open GL** (Open Graphcis Library). Conjunto de especificaciones estándar que definen una API multilenguaje y multiplataforma para escribir aplicaciones o juegos que producen gráficos en 3D.
- **DirectX**. Se trata de un conjunto de APIs como en el caso de OpenGL pero para la plataforma Microsoft Windows.
- **Plugins**. Aplicación adicional que se integra en otra con el fin de ofrecerle una función nueva y específica.
- **Open Source**. Software distribuido y desarrollado libremente.
- **Frames**. Cada uno de los fotogramas en los que se divide un video o una animación.

- **Streaming.** Técnica mediante la cual se permite reproducir contenidos multimedia sin necesidad de ser descargados previamente, almacenándolos en un buffer.
- **Middleware.** Se trata de un software capaz de asistir a otra aplicación con el fin de que interactúe con otras aplicaciones, software, redes, hardware o sistemas operativos.
- **DLL** (Dynamic Linking Library). Es la implementación de Microsoft del concepto de bibliotecas (librerías) compartidas en sistemas Windows y OS/2.

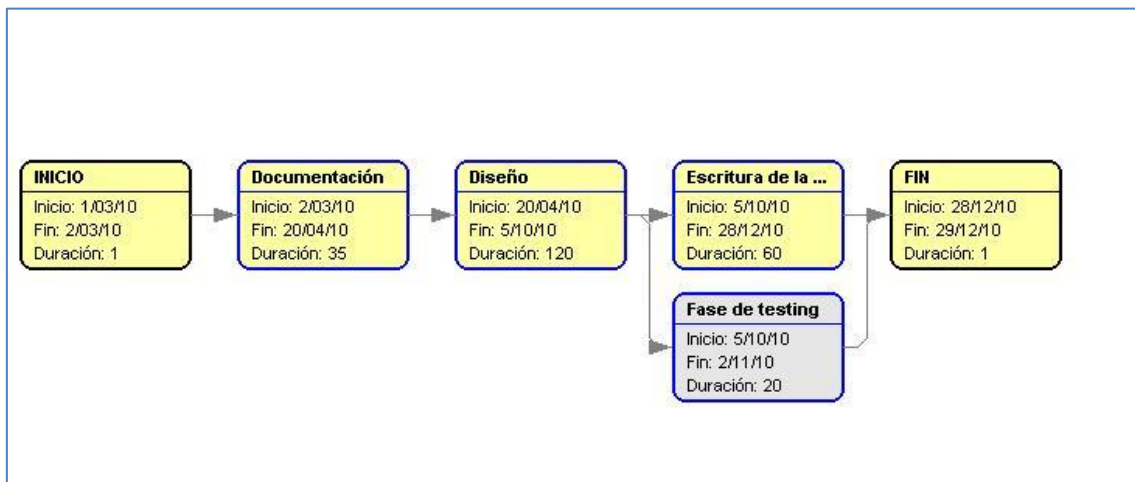
ANEXO II. Diagrama de GANTT inicial.



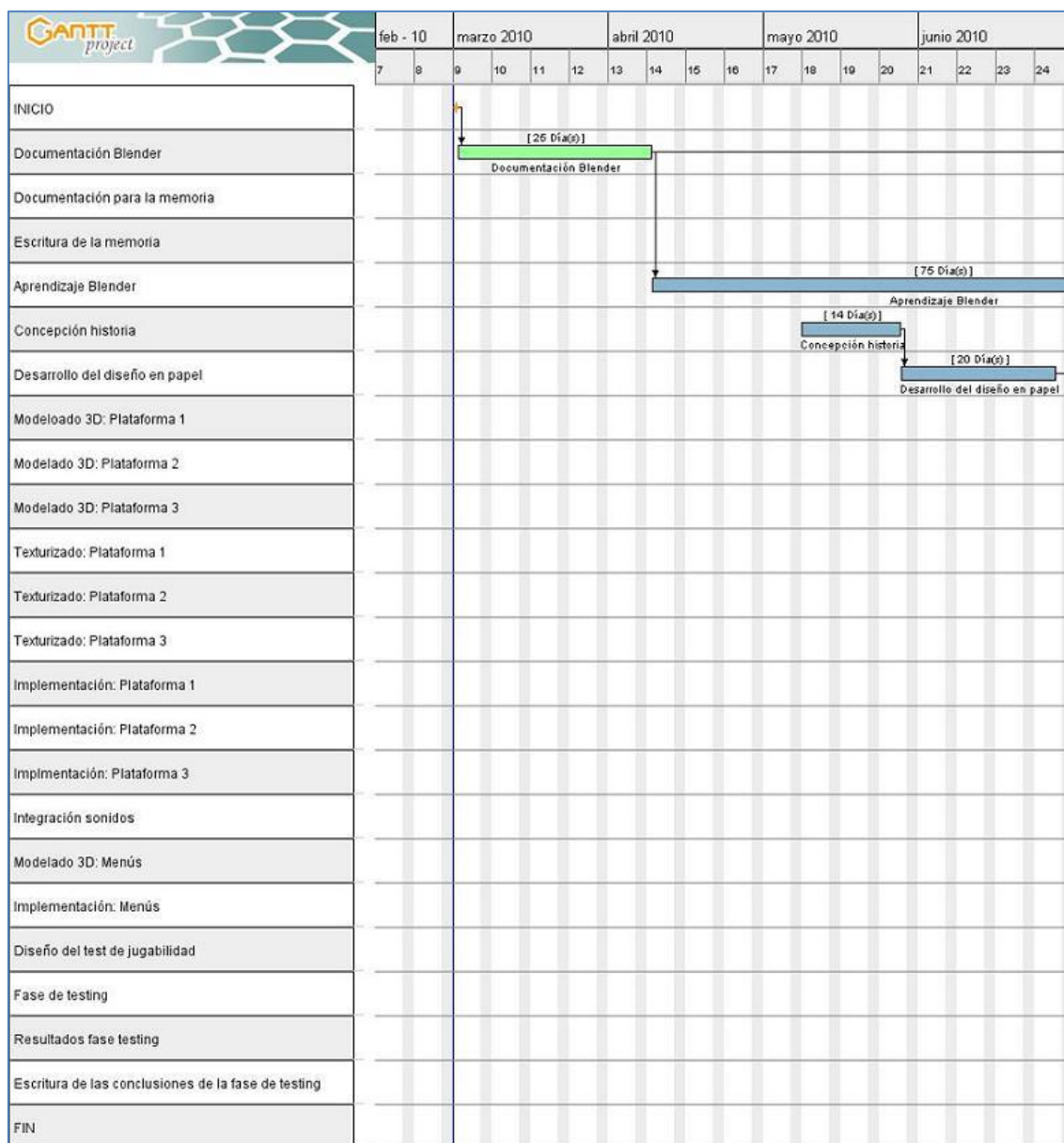


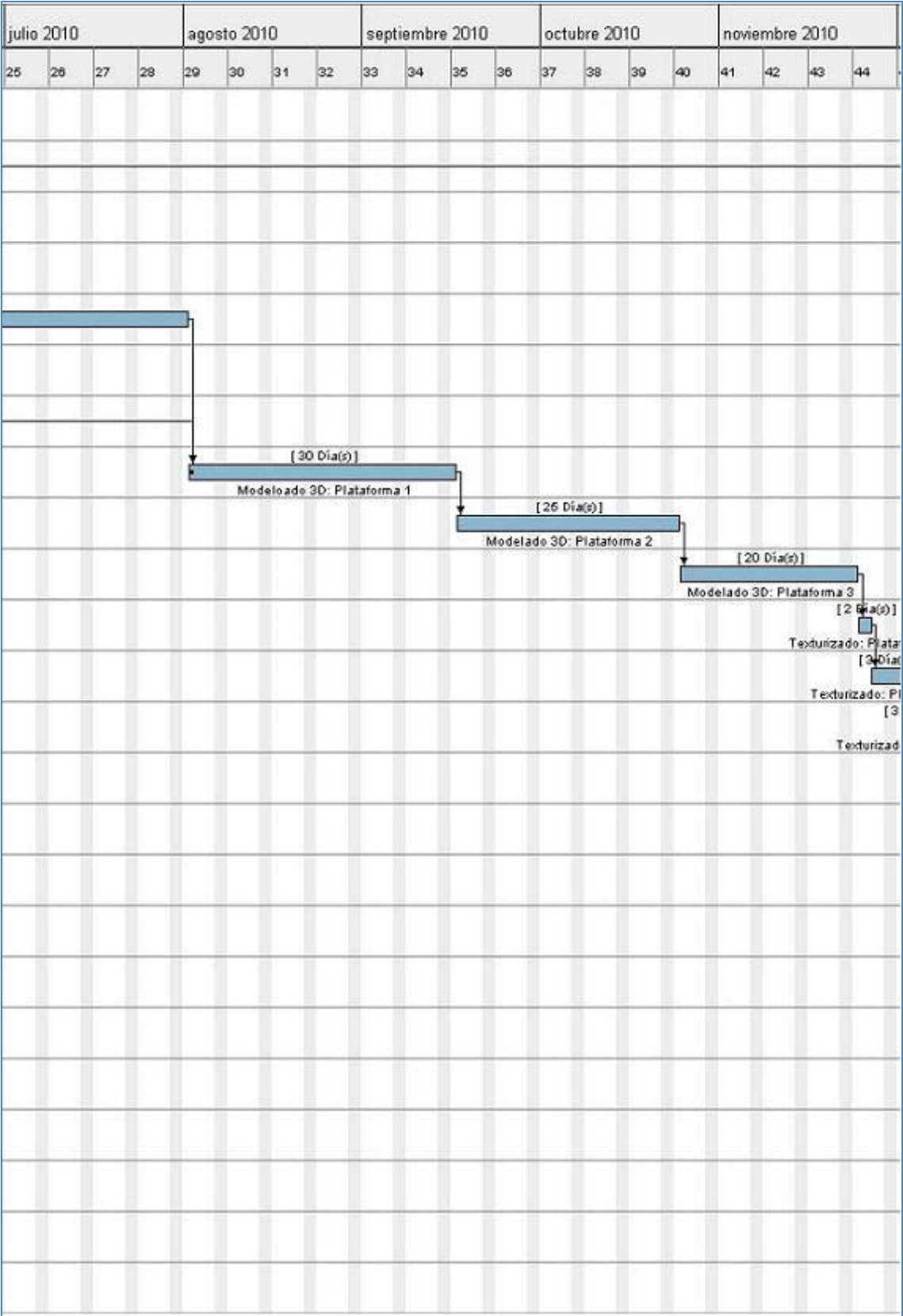


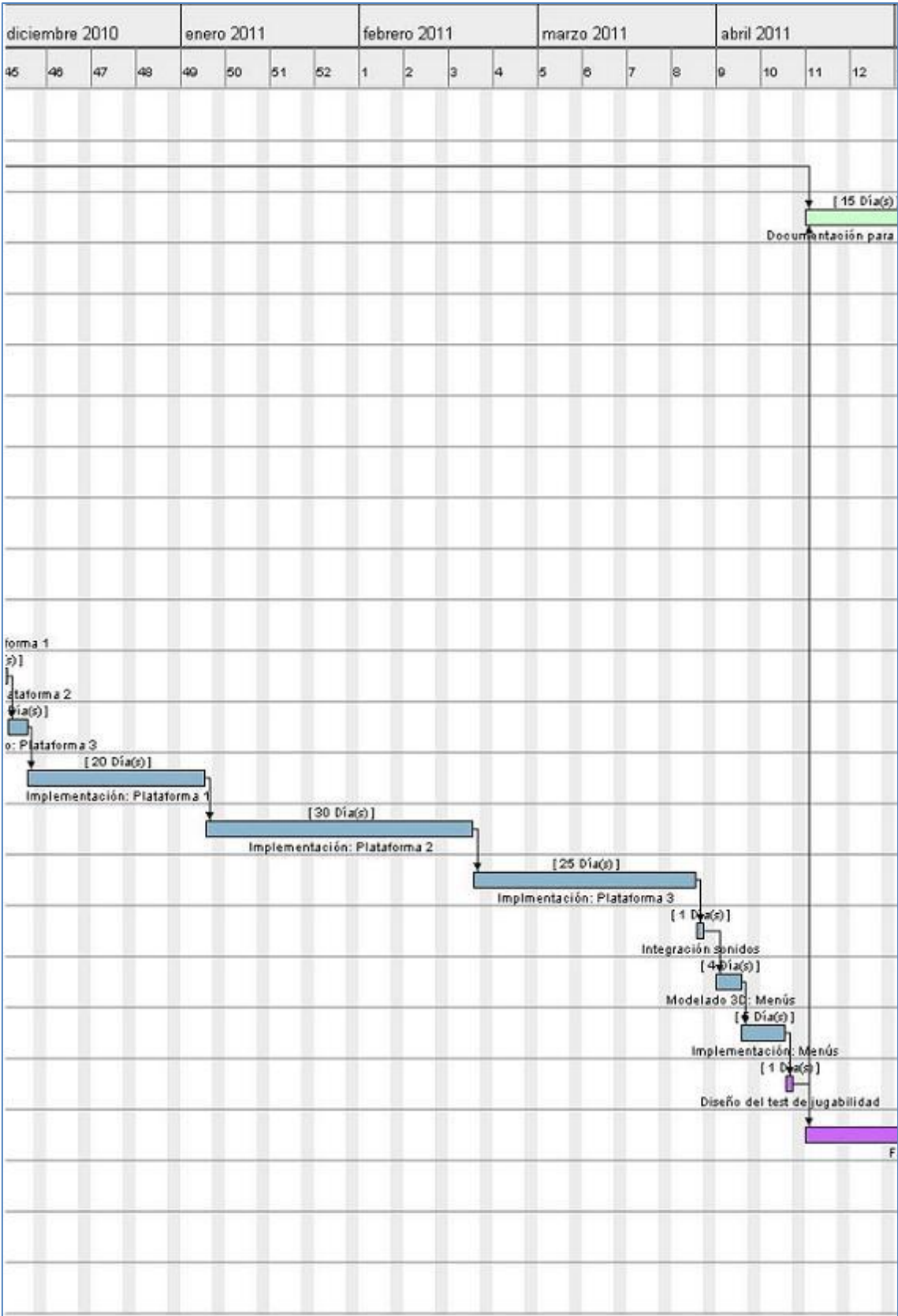
ANEXO III. Diagrama de PERT inicial.

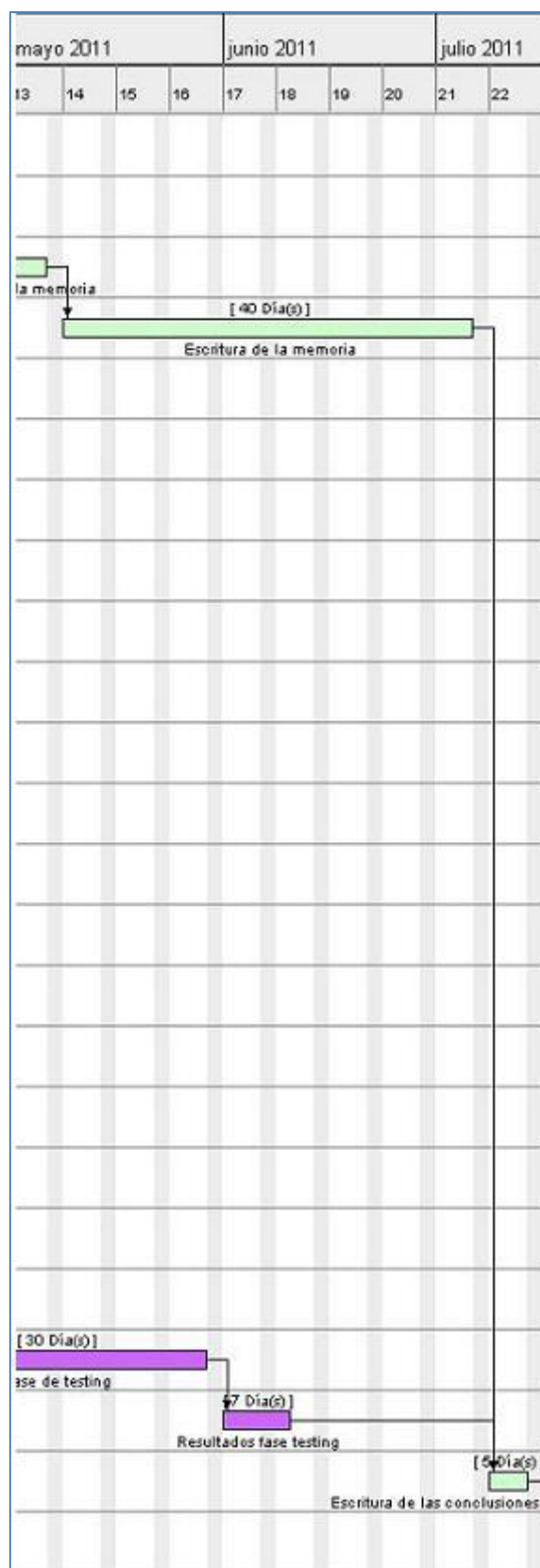


ANEXO IV. Diagrama de GANTT final.

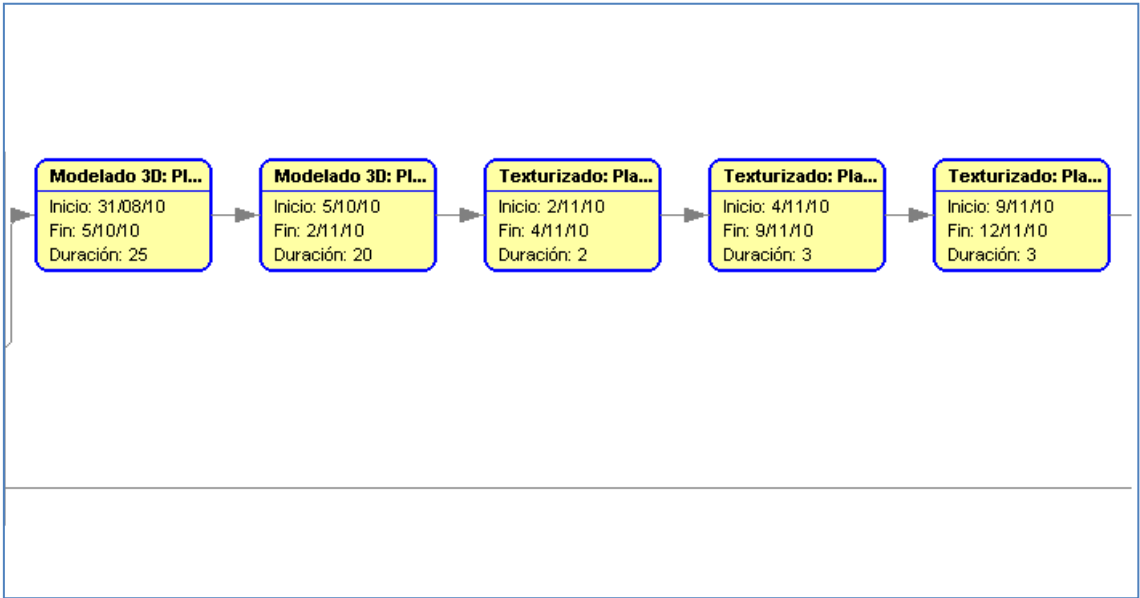
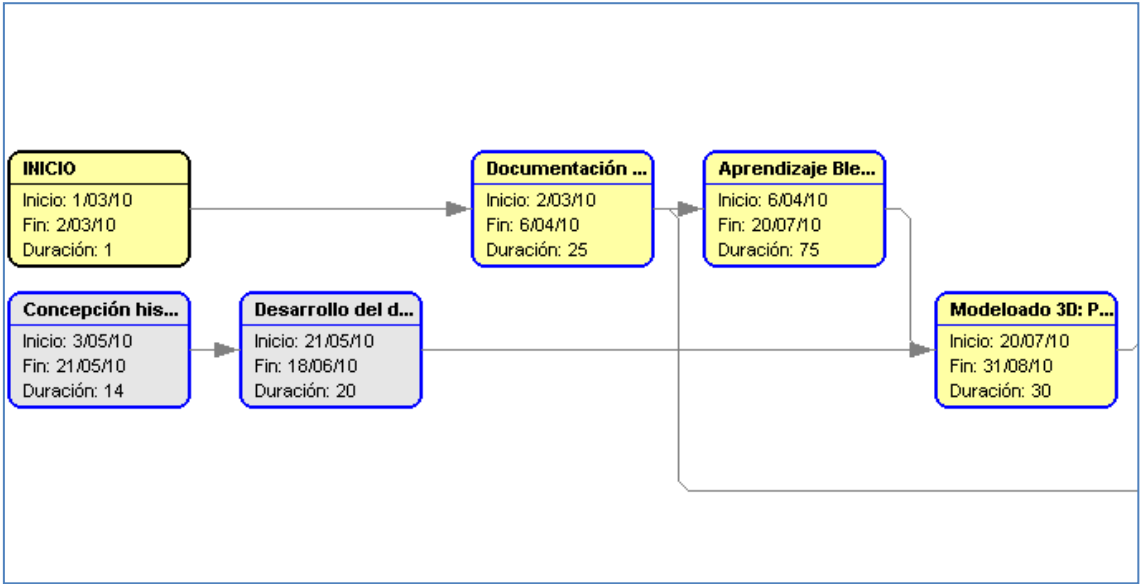


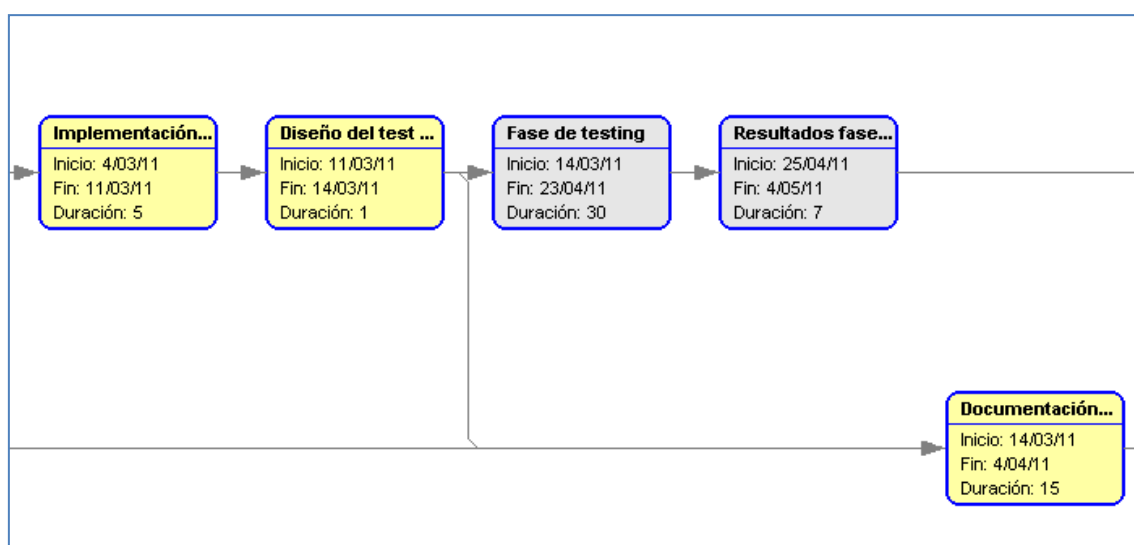
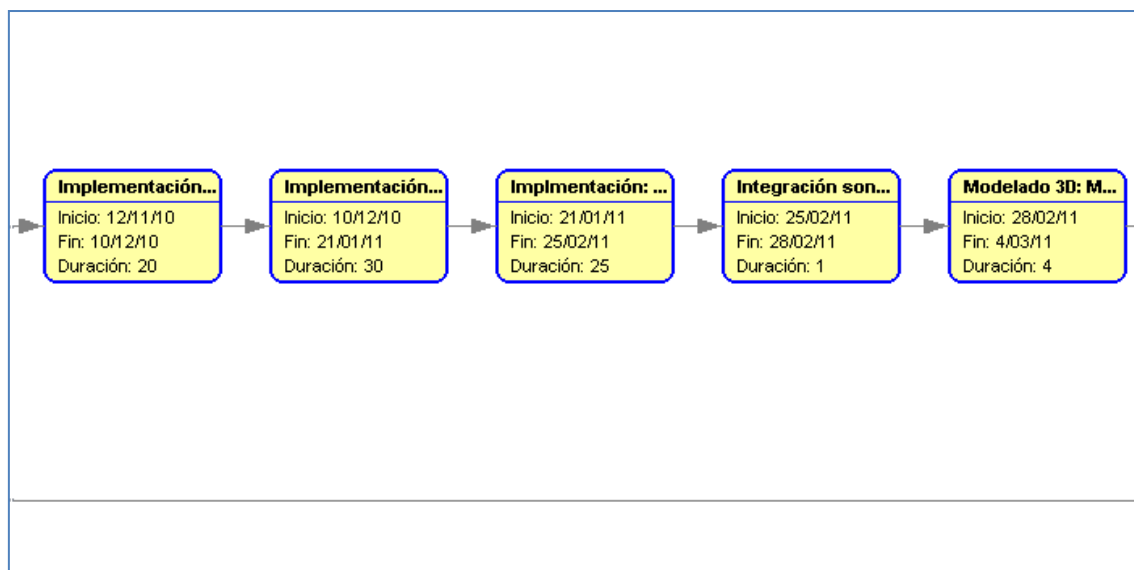


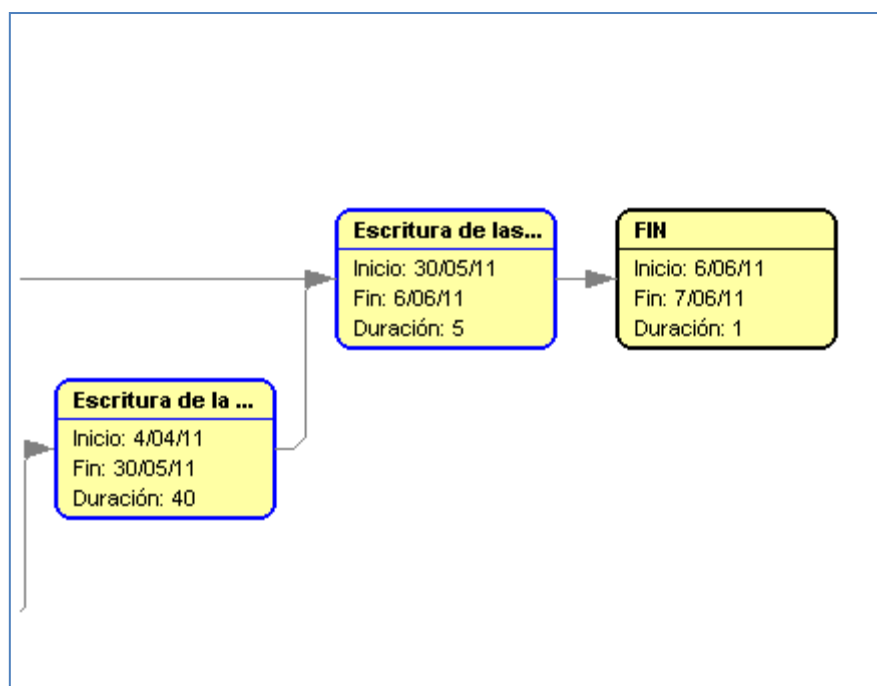




ANEXO V. Diagrama de PERT final.







ANEXO VI. Test de jugabilidad.

TEST DE EVALUACIÓN DEL VIDEOJUEGO

Evalúa de 1 (muy poco) a 5 (mucho) los siguientes aspectos del juego. Debes ser totalmente sincero, gracias a tus respuestas se hará un análisis eficiente de videojuego.

ANÁLISIS DE LA JUGABILIDAD

SATISFACCIÓN

1. Encuentras divertido el juego:

- ☒ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

APRENDIZAJE

1. Has jugado alguna vez a un juego parecido:

- ☒ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

2. Te ha sido difícil comprender la mecánica del juego:

- ☒ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

3. Crees que la dificultad está bien ajustada a cada nivel:

- ☒ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

4. Has sentido que has sido penalizado repetidamente por el mismo error:

- ☒ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

EFFECTIVIDAD

1. Ha motivado que lo hayas completado hasta el final:

- ☒ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

2. Lo has encontrado corto de duración:

- ☒ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

INMERSIÓN

1. Ha hecho que mantengas la atención durante todo el juego:

- ☒ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5



2. Estéticamente lo ves acertado:

- ☒ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

3. Los gráficos son agradables a la vista:

- ☒ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

4. Crees que el tiempo es suficiente para llegar a la meta:

- ☒ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

MOTIVACIÓN

1. Te ha hecho motivarte para conseguir llegar al final:

- ☒ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

EMOCIÓN

1. Te han gustado los efectos sonoros:

- ☒ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

2. Te ha despertado emoción (tensión, alegría, frustración...) mientras jugabas:

- ☒ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

ANÁLISIS PERSONAL

1. ¿Qué es lo que más te ha gustado?

2. ¿Qué es lo que menos te ha gustado?

3. ¿Qué mejorarías?



4. ¿Qué nombre le pondrías al juego?

5. Valoración general

Haga click para cambiar su valor.



ANEXO VII. Tests del grupo de menores de 30 años.

Álvaro Pujol Rodríguez, 24 años.

TEST DE EVALUACIÓN DEL VIDEOJUEGO

Evalúa de 1 (muy poco) a 5 (mucho) los siguientes aspectos del juego. Debes ser totalmente sincero, gracias a tus respuestas se hará un análisis eficiente de videojuego.

ANÁLISIS DE LA JUGABILIDAD

SATISFACCIÓN

1. Encuentras divertido el juego:

☐ 1 ☐ 2 ☐ 3 ☒ 4 ☐ 5

APRENDIZAJE

1. Has jugado alguna vez a un juego parecido:

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☒ 5

2. Te ha sido difícil comprender la mecánica del juego:

☐ 1 ☒ 2 ☐ 3 ☐ 4 ☐ 5

3. Crees que la dificultad está bien ajustada a cada nivel:

☐ 1 ☐ 2 ☒ 3 ☐ 4 ☐ 5

4. Has sentido que has sido penalizado repetidamente por el mismo error:

☒ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

EFFECTIVIDAD

1. Ha motivado que lo hayas completado hasta el final:

☐ 1 ☐ 2 ☐ 3 ☒ 4 ☐ 5

2. Lo has encontrado corto de duración:

☐ 1 ☐ 2 ☐ 3 ☒ 4 ☐ 5

INMERSIÓN

1. Ha hecho que mantengas la atención durante todo el juego:

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☒ 5

2. Estéticamente lo ves acertado:

☐ 1 ☐ 2 ☒ 3 ☐ 4 ☐ 5

3. Los gráficos son agradables a la vista:

☐ 1 ☐ 2 ☒ 3 ☐ 4 ☐ 5

4. Crees que el tiempo es suficiente para llegar a la meta:

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☒ 5

MOTIVACIÓN

1. Te ha hecho motivarte para conseguir llegar al final:

☐ 1 ☐ 2 ☐ 3 ☒ 4 ☐ 5

EMOCIÓN

1. Te han gustado los efectos sonoros:

☐ 1 ☐ 2 ☒ 3 ☐ 4 ☐ 5

2. Te ha despertado emoción (tensión, alegría, frustración...) mientras jugabas:

☐ 1 ☐ 2 ☐ 3 ☒ 4 ☐ 5

ANÁLISIS PERSONAL

1. ¿Qué es lo que más te ha gustado?

Los distintos tipos de plataformas.

2. ¿Qué es lo que menos te ha gustado?

La velocidad de reacción.

3. ¿Qué mejorarías?

Añadir mas niveles.

4. ¿Qué nombre le pondrías al juego?

Bolita pelotera

5. Valoración general

7

Clara Parra Gil, 24 años.

TEST DE EVALUACIÓN DEL VIDEOJUEGO

Evalúa de 1 (muy poco) a 5 (mucho) los siguientes aspectos del juego. Debes ser totalmente sincero, gracias a tus respuestas se hará un análisis eficiente de videojuego.

ANÁLISIS DE LA JUGABILIDAD

SATISFACCIÓN

1. Encuentras divertido el juego:

2. Te ha sido difícil comprender la mecánica del juego:

3. Crees que la dificultad está bien ajustada a cada nivel:

4. Has sentido que has sido penalizado repetidamente por el mismo error:

EFFECTIVIDAD

1. Ha motivado que lo hayas completado hasta el final:

2. Lo has encontrado corto de duración:

INMERSIÓN

1. Ha hecho que mantengas la atención durante todo el juego:

2. Estéticamente lo ves acertado:

3. Los gráficos son agradables a la vista:

4. Crees que el tiempo es suficiente para llegar a la meta:

MOTIVACIÓN

1. Te ha hecho motivarte para conseguir llegar al final:

EMOCIÓN

1. Te ha gustado los efectos sonoros:

2. Te ha despertado emoción (tensión, alegría, frustración...) mientras jugabas:

ANÁLISIS PERSONAL

1. ¿Qué es lo que más te ha gustado?

Los colores me parecían muy llamativos. Las tres pantallas eran muy entretenidas.

2. ¿Qué es lo que menos te ha gustado?

La gravedad.

3. ¿Qué mejorarías?

El tiempo de espera.

4. ¿Qué nombre le pondrías al juego?

superjuego ana

5. Valoración general

7

Daniel Rodríguez, 18 años.

TEST DE EVALUACIÓN DEL VIDEOJUEGO

Evalúa de 1 (muy poco) a 5 (mucho) los siguientes aspectos del juego. Debes ser totalmente sincero, gracias a tus respuestas se hará un análisis eficiente de videojuego.

ANÁLISIS DE LA JUGABILIDAD

SATISFACCIÓN

1. Encuentras divertido el juego:

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☒ 5

APRENDIZAJE

1. Has jugado alguna vez a un juego parecido:

☐ 1 ☐ 2 ☒ 3 ☐ 4 ☐ 5

2. Te ha sido difícil comprender la mecánica del juego:

☐ 1 ☒ 2 ☐ 3 ☐ 4 ☐ 5

3. Crees que la dificultad está bien ajustada a cada nivel:

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☒ 5

4. Has sentido que has sido penalizado repetidamente por el mismo error:

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☒ 5

EFFECTIVIDAD

1. Ha motivado que lo hayas completado hasta el final:

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☒ 5

2. Lo has encontrado corto de duración:

☐ 1 ☒ 2 ☐ 3 ☐ 4 ☐ 5

INMERSIÓN

1. Ha hecho que mantengas la atención durante todo el juego:

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☒ 5

2. Estéticamente lo ves acertado:

☐ 1 ☐ 2 ☐ 3 ☒ 4 ☐ 5

3. Los gráficos son agradables a la vista:

☐ 1 ☐ 2 ☐ 3 ☒ 4 ☐ 5

4. Crees que el tiempo es suficiente para llegar a la meta:

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☒ 5

MOTIVACIÓN

1. Te ha hecho motivarte para conseguir llegar al final:

☒ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

EMOCIÓN

1. Te han gustado los efectos sonoros:

☐ 1 ☐ 2 ☐ 3 ☒ 4 ☐ 5

2. Te ha despertado emoción (tensión, alegría, frustración...) mientras jugabas:

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☒ 5

ANÁLISIS PERSONAL

1. ¿Qué es lo que más te ha gustado?

Que me he caído más en la primera pantalla que en las demás.

2. ¿Qué es lo que menos te ha gustado?

Algunos pequeños errores del juego.

3. ¿Qué mejorarías?

Los gráficos.

4. ¿Qué nombre le pondrías al juego?

BoonBall

5. Valoración general

8

Elsa Íñigo Casado, 24 años.

TEST DE EVALUACIÓN DEL VIDEOJUEGO

Evalúa de 1 (muy poco) a 5 (mucho) los siguientes aspectos del juego. Debes ser totalmente sincero, gracias a tus respuestas se hará un análisis eficiente de videojuego.

ANÁLISIS DE LA JUGABILIDAD

SATISFACCIÓN

1. Encuentras divertido el juego:

APRENDIZAJE

1. Has jugado alguna vez a un juego parecido:

2. Te ha sido difícil comprender la mecánica del juego:

3. Crees que la dificultad está bien ajustada a cada nivel:

4. Has sentido que has sido penalizado repetidamente por el mismo error:

EFFECTIVIDAD

1. Ha motivado que lo hayas completado hasta el final:

2. Lo has encontrado corto de duración:

INMERSIÓN

1. Ha hecho que mantengas la atención durante todo el juego:

2. Estéticamente lo ves acertado:

3. Los gráficos son agradables a la vista:

4. Crees que el tiempo es suficiente para llegar a la meta:

MOTIVACIÓN

1. Te ha hecho motivarte para conseguir llegar al final:

EMOCIÓN

1. Te han gustado los efectos sonoros:

2. Te ha despertado emoción (tensión, alegría, frustración...) mientras jugabas:

ANÁLISIS PERSONAL

1. ¿Qué es lo que más te ha gustado?

El tipo de juego

2. ¿Qué es lo que menos te ha gustado?

Los colores del juego

3. ¿Qué mejorarías?

Los colores y las cámaras

4. ¿Qué nombre le pondrías al juego?

Super boing

5. Valoración general

9

Jorge Acebes Sánchez, 24 años.

TEST DE EVALUACIÓN DEL VIDEOJUEGO

Evalúa de 1 (muy poco) a 5 (mucho) los siguientes aspectos del juego. Debes ser totalmente sincero, gracias a tus respuestas se hará un análisis eficiente de videojuego.

ANÁLISIS DE LA JUGABILIDAD

SATISFACCIÓN

1. Encuentras divertido el juego:
- ☐ 1 ☐ 2 ☐ 3 ☒ 4 ☐ 5

APRENDIZAJE

1. Has jugado alguna vez a un juego parecido:
- ☐ 1 ☐ 2 ☒ 3 ☐ 4 ☐ 5
2. Te ha sido difícil comprender la mecánica del juego:
- ☐ 1 ☐ 2 ☐ 3 ☒ 4 ☐ 5
3. Crees que la dificultad está bien ajustada a cada nivel:
- ☐ 1 ☐ 2 ☒ 3 ☐ 4 ☐ 5
4. Has sentido que has sido penalizado repetidamente por el mismo error:
- ☐ 1 ☐ 2 ☒ 3 ☐ 4 ☐ 5

EFFECTIVIDAD

1. Ha motivado que lo hayas completado hasta el final:
- ☐ 1 ☐ 2 ☒ 3 ☐ 4 ☐ 5
2. Lo has encontrado corto de duración:
- ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☒ 5

INMERSIÓN

1. Ha hecho que mantengas la atención durante todo el juego:
- ☐ 1 ☒ 2 ☐ 3 ☐ 4 ☐ 5

2. Estéticamente lo ves acertado:
- ☐ 1 ☒ 2 ☐ 3 ☐ 4 ☐ 5
3. Los gráficos son agradables a la vista:
- ☐ 1 ☐ 2 ☒ 3 ☐ 4 ☐ 5
4. Crees que el tiempo es suficiente para llegar a la meta:
- ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☒ 5

MOTIVACIÓN

1. Te ha hecho motivarte para conseguir llegar al final:
- ☐ 1 ☐ 2 ☐ 3 ☒ 4 ☐ 5

EMOCIÓN

1. Te han gustado los efectos sonoros:
- ☐ 1 ☒ 2 ☐ 3 ☐ 4 ☐ 5
2. Te ha despertado emoción (tensión, alegría, frustración...) mientras jugabas:
- ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☒ 5

ANÁLISIS PERSONAL

1. ¿Qué es lo que más te ha gustado?

La variabilidad de movimientos.

2. ¿Qué es lo que menos te ha gustado?

Las placas fantasma sin aviso.

3. ¿Qué mejorarías?

Poder tocar dos teclas a la vez sin fallo(diagonal).
Más niveles.
Explicaciones de los niveles y las posibilidades del juego.
Los relieves del final.

4. ¿Qué nombre le pondrías al juego?

Bolatron.

5. Valoración general
- 7

José Luis Montiel Muñoz, 24 años.

TEST DE EVALUACIÓN DEL VIDEOJUEGO

Evalúa de 1 (muy poco) a 5 (mucho) los siguientes aspectos del juego. Debes ser totalmente sincero, gracias a tus respuestas se hará un análisis eficiente de videojuego.

ANÁLISIS DE LA JUGABILIDAD

SATISFACCIÓN

1. Encuentras divertido el juego:

1

2

3

4

5

APRENDIZAJE

1. Has jugado alguna vez a un juego parecido:

1

2

3

4

5

2. Te ha sido difícil comprender la mecánica del juego:

1

2

3

4

5

3. Crees que la dificultad está bien ajustada a cada nivel:

1

2

3

4

5

4. Has sentido que has sido penalizado repetidamente por el mismo error:

1

2

3

4

5

EFFECTIVIDAD

1. Ha motivado que lo hayas completado hasta el final:

1

2

3

4

5

2. Lo has encontrado corto de duración:

1

2

3

4

5

INMERSIÓN

1. Ha hecho que mantengas la atención durante todo el juego:

1

2

3

4

5

2. Estéticamente lo ves acertado:

1

2

3

4

5

3. Los gráficos son agradables a la vista:

1

2

3

4

5

4. Crees que el tiempo es suficiente para llegar a la meta:

1

2

3

4

5

MOTIVACIÓN

1. Te ha hecho motivarte para conseguir llegar al final:

1

2

3

4

5

EMOCIÓN

1. Te han gustado los efectos sonoros:

1

2

3

4

5

2. Te ha despertado emoción (tensión, alegría, frustración...) mientras jugabas:

1

2

3

4

5

ANÁLISIS PERSONAL

1. ¿Qué es lo que más te ha gustado?

el colorido

2. ¿Qué es lo que menos te ha gustado?

Los efectos sonoros

3. ¿Qué mejorarías?

el control

4. ¿Qué nombre le pondrías al juego?

mortal ball

5. Valoración general

6

Rafael Román Rodríguez, 19 años.

TEST DE EVALUACIÓN DEL VIDEOJUEGO

Evalúa de 1 (muy poco) a 5 (mucho) los siguientes aspectos del juego. Debes ser totalmente sincero, gracias a tus respuestas se hará un análisis eficiente de videojuego.

ANÁLISIS DE LA JUGABILIDAD

SATISFACCIÓN

1. Encuentras divertido el juego:

1

2

3

4

5

APRENDIZAJE

1. Has jugado alguna vez a un juego parecido:

1

2

3

4

5

2. Te ha sido difícil comprender la mecánica del juego:

1

2

3

4

5

3. Crees que la dificultad está bien ajustada a cada nivel:

1

2

3

4

5

4. Has sentido que has sido penalizado repetidamente por el mismo error:

1

2

3

4

5

EFFECTIVIDAD

1. Ha motivado que lo hayas completado hasta el final:

1

2

3

4

5

2. Lo has encontrado corto de duración:

1

2

3

4

5

INMERSIÓN

1. Ha hecho que mantengas la atención durante todo el juego:

1

2

3

4

5

2. Estéticamente lo ves acertado:

1

2

3

4

5

3. Los gráficos son agradables a la vista:

1

2

3

4

5

4. Crees que el tiempo es suficiente para llegar a la meta:

1

2

3

4

5

MOTIVACIÓN

1. Te ha hecho motivarte para conseguir llegar al final:

1

2

3

4

5

EMOCIÓN

1. Te han gustado los efectos sonoros:

1

2

3

4

5

2. Te ha despertado emoción (tensión, alegría, frustración...) mientras jugabas:

1

2

3

4

5

ANÁLISIS PERSONAL

1. ¿Qué es lo que más te ha gustado?

Los efectos sonoros.

2. ¿Qué es lo que menos te ha gustado?

La dificultad de la primera pantalla es mayor a la de la segunda

3. ¿Qué mejorarías?

La dificultad y los retos.

4. ¿Qué nombre le pondrías al juego?

UltraBall

5. Valoración general

8

176

Jaime Recarte Roldán, 23 años.

TEST DE EVALUACIÓN DEL VIDEOJUEGO

Evalúa de 1 (muy poco) a 5 (mucho) los siguientes aspectos del juego. Debes ser totalmente sincero, gracias a tus respuestas se hará un análisis eficiente de videojuego.

ANÁLISIS DE LA JUGABILIDAD

SATISFACCIÓN

1. Encuentras divertido el juego:

APRENDIZAJE

1. Has jugado alguna vez a un juego parecido:

2. Te ha sido difícil comprender la mecánica del juego:

3. Crees que la dificultad está bien ajustada a cada nivel:

4. Has sentido que has sido penalizado repetidamente por el mismo error:

EFFECTIVIDAD

1. Ha motivado que lo hayas completado hasta el final:

2. Lo has encontrado corto de duración:

INMERSIÓN

1. Ha hecho que mantengas la atención durante todo el juego:

2. Estéticamente lo ves acertado:

3. Los gráficos son agradables a la vista:

4. Crees que el tiempo es suficiente para llegar a la meta:

MOTIVACIÓN

1. Te ha hecho motivarte para conseguir llegar al final:

EMOCIÓN

1. Te han gustado los efectos sonoros:

2. Te ha despertado emoción (tensión, alegría, frustración...) mientras jugabas:

ANÁLISIS PERSONAL

1. ¿Qué es lo que más te ha gustado?

La emoción

2. ¿Qué es lo que menos te ha gustado?

El sonido de caída

3. ¿Qué mejorarías?

Los gráficos

4. ¿Qué nombre le pondrías al juego?

TouchingBalls

5. Valoración general

8

Gabriel Rodríguez Mozos, 21 años.

TEST DE EVALUACIÓN DEL VIDEOJUEGO

Evalúa de 1 (muy poco) a 5 (mucho) los siguientes aspectos del juego. Debes ser totalmente sincero, gracias a tus respuestas se hará un análisis eficiente de videojuego.

ANÁLISIS DE LA JUGABILIDAD

SATISFACCIÓN

1. Encuentras divertido el juego:

☐ 1 ☐ 2 ☒ 3 ☐ 4 ☐ 5

APRENDIZAJE

1. Has jugado alguna vez a un juego parecido:

☐ 1 ☐ 2 ☒ 3 ☐ 4 ☐ 5

2. Te ha sido difícil comprender la mecánica del juego:

☐ 1 ☒ 2 ☐ 3 ☐ 4 ☐ 5

3. Crees que la dificultad está bien ajustada a cada nivel:

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☒ 5

4. Has sentido que has sido penalizado repetidamente por el mismo error:

☐ 1 ☐ 2 ☒ 3 ☐ 4 ☐ 5

EFFECTIVIDAD

1. Ha motivado que lo hayas completado hasta el final:

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☒ 5

2. Lo has encontrado corto de duración:

☒ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

INMERSIÓN

1. Ha hecho que mantengas la atención durante todo el juego:

☐ 1 ☐ 2 ☐ 3 ☒ 4 ☐ 5

2. Estéticamente lo ves acertado:

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☒ 5

3. Los gráficos son agradables a la vista:

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☒ 5

4. Crees que el tiempo es suficiente para llegar a la meta:

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☒ 5

MOTIVACIÓN

1. Te ha hecho motivarte para conseguir llegar al final:

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☒ 5

EMOCIÓN

1. Te han gustado los efectos sonoros:

☐ 1 ☐ 2 ☐ 3 ☒ 4 ☐ 5

2. Te ha despertado emoción (tensión, alegría, frustración...) mientras jugabas:

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☒ 5

ANÁLISIS PERSONAL

1. ¿Qué es lo que más te ha gustado?

La jugabilidad

2. ¿Qué es lo que menos te ha gustado?

El cambio de las camaras

3. ¿Qué mejorarías?

Los pixels

4. ¿Qué nombre le pondrías al juego?

Bolita tontita

5. Valoración general

7

Victoria González Domínguez, 24 años.

TEST DE EVALUACIÓN DEL VIDEOJUEGO

Evalúa de 1 (muy poco) a 5 (mucho) los siguientes aspectos del juego. Debes ser totalmente sincero, gracias a tus respuestas se hará un análisis eficiente de videojuego.

ANÁLISIS DE LA JUGABILIDAD

SATISFACCIÓN

1. Encuentras divertido el juego:

☐ 1 ☐ 2 ☒ 3 ☐ 4 ☐ 5

APRENDIZAJE

1. Has jugado alguna vez a un juego parecido:

☐ 1 ☐ 2 ☐ 3 ☒ 4 ☐ 5

2. Te ha sido difícil comprender la mecánica del juego:

☐ 1 ☐ 2 ☒ 3 ☐ 4 ☐ 5

3. Crees que la dificultad está bien ajustada a cada nivel:

☐ 1 ☐ 2 ☐ 3 ☒ 4 ☐ 5

4. Has sentido que has sido penalizado repetidamente por el mismo error:

☐ 1 ☐ 2 ☐ 3 ☒ 4 ☐ 5

EFECTIVIDAD

1. Ha motivado que lo hayas completado hasta el final:

☐ 1 ☐ 2 ☒ 3 ☐ 4 ☐ 5

2. Lo has encontrado corto de duración:

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☒ 5

INMERSIÓN

1. Ha hecho que mantengas la atención durante todo el juego:

☐ 1 ☐ 2 ☒ 3 ☐ 4 ☐ 5

2. Estéticamente lo ves acertado:

☐ 1 ☐ 2 ☐ 3 ☒ 4 ☐ 5

3. Los gráficos son agradables a la vista:

☐ 1 ☐ 2 ☐ 3 ☒ 4 ☐ 5

4. Crees que el tiempo es suficiente para llegar a la meta:

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☒ 5

MOTIVACIÓN

1. Te ha hecho motivarte para conseguir llegar al final:

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☒ 5

EMOCIÓN

1. Te han gustado los efectos sonoros:

☐ 1 ☐ 2 ☒ 3 ☐ 4 ☐ 5

2. Te ha despertado emoción (tensión, alegría, frustración...) mientras jugabas:

☐ 1 ☐ 2 ☒ 3 ☐ 4 ☐ 5

ANÁLISIS PERSONAL

1. ¿Qué es lo que más te ha gustado?

Que cada vez hay que hacer cosas distintas en los niveles

2. ¿Qué es lo que menos te ha gustado?

Falta de explicaciones

3. ¿Qué mejorarías?

La estética

4. ¿Qué nombre le pondrías al juego?

Bolliing

5. Valoración general

7

ANEXO VIII. Tests del grupo de mayores de 30 años.

Antonio Rodríguez Sánchez, 45 años.

TEST DE EVALUACIÓN DEL VIDEOJUEGO	
Evalúa de 1 (muy poco) a 5 (mucho) los siguientes aspectos del juego. Debes ser totalmente sincero, gracias a tus respuestas se hará un análisis eficiente de videojuego.	
ANÁLISIS DE LA JUGABILIDAD	
SATISFACCIÓN	
1. Encuentras divertido el juego: <input type="radio"/> 1 <input type="radio"/> 2 <input type="radio"/> 3 <input checked="" type="radio"/> 4 <input type="radio"/> 5	
APRENDIZAJE	
1. Has jugado alguna vez a un juego parecido: <input checked="" type="radio"/> 1 <input type="radio"/> 2 <input type="radio"/> 3 <input type="radio"/> 4 <input type="radio"/> 5	
2. Te ha sido difícil comprender la mecánica del juego: <input type="radio"/> 1 <input checked="" type="radio"/> 2 <input type="radio"/> 3 <input type="radio"/> 4 <input type="radio"/> 5	
3. Crees que la dificultad está bien ajustada a cada nivel: <input type="radio"/> 1 <input type="radio"/> 2 <input type="radio"/> 3 <input checked="" type="radio"/> 4 <input type="radio"/> 5	
4. Has sentido que has sido penalizado repetidamente por el mismo error: <input type="radio"/> 1 <input type="radio"/> 2 <input type="radio"/> 3 <input checked="" type="radio"/> 4 <input type="radio"/> 5	
EFFECTIVIDAD	
1. Ha motivado que lo hayas completado hasta el final: <input type="radio"/> 1 <input checked="" type="radio"/> 2 <input type="radio"/> 3 <input type="radio"/> 4 <input type="radio"/> 5	
2. Lo has encontrado corto de duración: <input type="radio"/> 1 <input type="radio"/> 2 <input type="radio"/> 3 <input type="radio"/> 4 <input checked="" type="radio"/> 5	
INMERSIÓN	
1. Ha hecho que mantengas la atención durante todo el juego: <input type="radio"/> 1 <input type="radio"/> 2 <input checked="" type="radio"/> 3 <input type="radio"/> 4 <input type="radio"/> 5	
2. Estéticamente lo ves acertado: <input type="radio"/> 1 <input type="radio"/> 2 <input type="radio"/> 3 <input checked="" type="radio"/> 4 <input type="radio"/> 5	
3. Los gráficos son agradables a la vista: <input type="radio"/> 1 <input type="radio"/> 2 <input type="radio"/> 3 <input checked="" type="radio"/> 4 <input type="radio"/> 5	
4. Crees que el tiempo es suficiente para llegar a la meta: <input type="radio"/> 1 <input type="radio"/> 2 <input type="radio"/> 3 <input type="radio"/> 4 <input checked="" type="radio"/> 5	
MOTIVACIÓN	
1. Te ha hecho motivarte para conseguir llegar al final: <input type="radio"/> 1 <input type="radio"/> 2 <input type="radio"/> 3 <input checked="" type="radio"/> 4 <input type="radio"/> 5	
EMOCIÓN	
1. Te han gustado los efectos sonoros: <input type="radio"/> 1 <input type="radio"/> 2 <input checked="" type="radio"/> 3 <input type="radio"/> 4 <input type="radio"/> 5	
2. Te ha despertado emoción (tensión, alegría, frustración...) mientras jugabas: <input type="radio"/> 1 <input type="radio"/> 2 <input type="radio"/> 3 <input type="radio"/> 4 <input checked="" type="radio"/> 5	
ANÁLISIS PERSONAL	
1. ¿Qué es lo que más te ha gustado? El efecto del cubo en la segunda pantalla	
2. ¿Qué es lo que menos te ha gustado? Me ha costado entender el objetivo del juego	
3. ¿Qué mejorarías? La explicación del juego	
4. ¿Qué nombre le pondrías al juego? Pelotita	
5. Valoración general 7	

Elvira Mozos, 59 años.

TEST DE EVALUACIÓN DEL VIDEOJUEGO	
<p>Evalúa de 1 (muy poco) a 5 (mucho) los siguientes aspectos del juego. Debes ser totalmente sincero, gracias a tus respuestas se hará un análisis eficiente de videojuego.</p> <p>ANÁLISIS DE LA JUGABILIDAD</p> <p>SATISFACCIÓN</p> <p>1. Encuentras divertido el juego:</p> <p><input type="radio"/> 1 <input type="radio"/> 2 <input type="radio"/> 3 <input type="radio"/> 4 <input checked="" type="radio"/> 5</p> <p>APRENDIZAJE</p> <p>1. Has jugado alguna vez a un juego parecido:</p> <p><input checked="" type="radio"/> 1 <input type="radio"/> 2 <input type="radio"/> 3 <input type="radio"/> 4 <input type="radio"/> 5</p> <p>2. Te ha sido difícil comprender la mecánica del juego:</p> <p><input checked="" type="radio"/> 1 <input type="radio"/> 2 <input type="radio"/> 3 <input type="radio"/> 4 <input type="radio"/> 5</p> <p>3. Crees que la dificultad está bien ajustada a cada nivel:</p> <p><input type="radio"/> 1 <input type="radio"/> 2 <input checked="" type="radio"/> 3 <input type="radio"/> 4 <input type="radio"/> 5</p> <p>4. Has sentido que has sido penalizado repetidamente por el mismo error:</p> <p><input type="radio"/> 1 <input type="radio"/> 2 <input checked="" type="radio"/> 3 <input type="radio"/> 4 <input type="radio"/> 5</p> <p>EFFECTIVIDAD</p> <p>1. Ha motivado que lo hayas completado hasta el final:</p> <p><input type="radio"/> 1 <input type="radio"/> 2 <input type="radio"/> 3 <input checked="" type="radio"/> 4 <input type="radio"/> 5</p> <p>2. Lo has encontrado corto de duración:</p> <p><input type="radio"/> 1 <input type="radio"/> 2 <input type="radio"/> 3 <input type="radio"/> 4 <input checked="" type="radio"/> 5</p> <p>INMERSIÓN</p> <p>1. Ha hecho que mantengas la atención durante todo el juego:</p> <p><input type="radio"/> 1 <input type="radio"/> 2 <input type="radio"/> 3 <input type="radio"/> 4 <input checked="" type="radio"/> 5</p>	
<p>2. Estéticamente lo ves acertado:</p> <p><input type="radio"/> 1 <input type="radio"/> 2 <input type="radio"/> 3 <input type="radio"/> 4 <input checked="" type="radio"/> 5</p> <p>3. Los gráficos son agradables a la vista:</p> <p><input type="radio"/> 1 <input type="radio"/> 2 <input checked="" type="radio"/> 3 <input type="radio"/> 4 <input type="radio"/> 5</p> <p>4. Crees que el tiempo es suficiente para llegar a la meta:</p> <p><input type="radio"/> 1 <input type="radio"/> 2 <input type="radio"/> 3 <input checked="" type="radio"/> 4 <input type="radio"/> 5</p> <p>MOTIVACIÓN</p> <p>1. Te ha hecho motivarte para conseguir llegar al final:</p> <p><input type="radio"/> 1 <input type="radio"/> 2 <input checked="" type="radio"/> 3 <input type="radio"/> 4 <input type="radio"/> 5</p> <p>EMOCIÓN</p> <p>1. Te han gustado los efectos sonoros:</p> <p><input type="radio"/> 1 <input type="radio"/> 2 <input type="radio"/> 3 <input checked="" type="radio"/> 4 <input type="radio"/> 5</p> <p>2. Te ha despertado emoción (tensión, alegría, frustración...) mientras jugabas:</p> <p><input type="radio"/> 1 <input type="radio"/> 2 <input type="radio"/> 3 <input checked="" type="radio"/> 4 <input type="radio"/> 5</p> <p>ANÁLISIS PERSONAL</p> <p>1. ¿Qué es lo que más te ha gustado?</p> <p>Los colores de los niveles</p> <p>2. ¿Qué es lo que menos te ha gustado?</p> <p>Al principio no entendía bien lo que tenía que hacer</p> <p>3. ¿Qué mejorarías?</p> <p>Pondría unas instrucciones más claras</p>	
<p>4. ¿Qué nombre le pondrías al juego?</p> <p>Anaball</p> <p>5. Valoración general</p> <p>7</p>	

Emilio García Cabeza, 32 años.

TEST DE EVALUACIÓN DEL VIDEOJUEGO	
Evalúa de 1 (muy poco) a 5 (mucho) los siguientes aspectos del juego. Debes ser totalmente sincero, gracias a tus respuestas se hará un análisis eficiente de videojuego.	
ANÁLISIS DE LA JUGABILIDAD	
SATISFACCIÓN	
1. Encuentras divertido el juego:	
<input type="radio"/> 1	<input type="radio"/> 2 <input type="radio"/> 3 <input checked="" type="radio"/> 4 <input type="radio"/> 5
APRENDIZAJE	
1. Has jugado alguna vez a un juego parecido:	
<input type="radio"/> 1	<input checked="" type="radio"/> 2 <input type="radio"/> 3 <input type="radio"/> 4 <input type="radio"/> 5
2. Te ha sido difícil comprender la mecánica del juego:	
<input type="radio"/> 1	<input type="radio"/> 2 <input checked="" type="radio"/> 3 <input type="radio"/> 4 <input type="radio"/> 5
3. Crees que la dificultad está bien ajustada a cada nivel:	
<input type="radio"/> 1	<input type="radio"/> 2 <input type="radio"/> 3 <input checked="" type="radio"/> 4 <input type="radio"/> 5
4. Has sentido que has sido penalizado repetidamente por el mismo error:	
<input type="radio"/> 1	<input type="radio"/> 2 <input checked="" type="radio"/> 3 <input type="radio"/> 4 <input type="radio"/> 5
EFECTIVIDAD	
1. Ha motivado que lo hayas completado hasta el final:	
<input checked="" type="radio"/> 1	<input type="radio"/> 2 <input type="radio"/> 3 <input type="radio"/> 4 <input type="radio"/> 5
2. Lo has encontrado corto de duración:	
<input type="radio"/> 1	<input type="radio"/> 2 <input type="radio"/> 3 <input type="radio"/> 4 <input checked="" type="radio"/> 5
INMERSIÓN	
1. Ha hecho que mantengas la atención durante todo el juego:	
<input type="radio"/> 1	<input type="radio"/> 2 <input type="radio"/> 3 <input checked="" type="radio"/> 4 <input type="radio"/> 5
2. Estéticamente lo ves acertado:	
<input type="radio"/> 1	<input type="radio"/> 2 <input type="radio"/> 3 <input checked="" type="radio"/> 4 <input type="radio"/> 5
3. Los gráficos son agradables a la vista:	
<input type="radio"/> 1	<input type="radio"/> 2 <input type="radio"/> 3 <input checked="" type="radio"/> 4 <input type="radio"/> 5
4. Crees que el tiempo es suficiente para llegar a la meta:	
<input type="radio"/> 1	<input type="radio"/> 2 <input checked="" type="radio"/> 3 <input type="radio"/> 4 <input type="radio"/> 5
MOTIVACIÓN	
1. Te ha hecho motivarte para conseguir llegar al final:	
<input type="radio"/> 1	<input checked="" type="radio"/> 2 <input type="radio"/> 3 <input type="radio"/> 4 <input type="radio"/> 5
EMOCIÓN	
1. Te han gustado los efectos sonoros:	
<input type="radio"/> 1	<input checked="" type="radio"/> 2 <input type="radio"/> 3 <input type="radio"/> 4 <input type="radio"/> 5
2. Te ha despertado emoción (tensión, alegría, frustración...) mientras jugabas:	
<input type="radio"/> 1	<input type="radio"/> 2 <input type="radio"/> 3 <input type="radio"/> 4 <input checked="" type="radio"/> 5
ANÁLISIS PERSONAL	
1. ¿Qué es lo que más te ha gustado?	
<div>La decoración</div>	
2. ¿Qué es lo que menos te ha gustado?	
<div>Me he aburrido un poco y los efectos sonoros no me han gustado</div>	
3. ¿Qué mejorarías?	
<div>Pondría más pantallas y mejoraría los retos.</div>	
4. ¿Qué nombre le pondrías al juego?	
<div>BALON_BOOM</div>	
5. Valoración general	
5	

Encarnación Rodríguez Sánchez, 53 años.

TEST DE EVALUACIÓN DEL VIDEOJUEGO

Evalúa de 1 (muy poco) a 5 (mucho) los siguientes aspectos del juego. Debes ser totalmente sincero, gracias a tus respuestas se hará un análisis eficiente de videojuego.

ANÁLISIS DE LA JUGABILIDAD

SATISFACCIÓN

1. Encuentras divertido el juego:

- ☐ 1 ☐ 2 ☒ 3 ☐ 4 ☐ 5

APRENDIZAJE

1. Has jugado alguna vez a un juego parecido:

- ☒ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

2. Te ha sido difícil comprender la mecánica del juego:

- ☐ 1 ☒ 2 ☐ 3 ☐ 4 ☐ 5

3. Crees que la dificultad está bien ajustada a cada nivel:

- ☐ 1 ☐ 2 ☐ 3 ☒ 4 ☐ 5

4. Has sentido que has sido penalizado repetidamente por el mismo error:

- ☐ 1 ☐ 2 ☐ 3 ☒ 4 ☐ 5

EFFECTIVIDAD

1. Ha motivado que lo hayas completado hasta el final:

- ☒ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

2. Lo has encontrado corto de duración:

- ☐ 1 ☒ 2 ☐ 3 ☐ 4 ☐ 5

INMERSIÓN

1. Ha hecho que mantengas la atención durante todo el juego:

- ☐ 1 ☐ 2 ☐ 3 ☒ 4 ☐ 5

2. Estéticamente lo ves acertado:

- ☐ 1 ☐ 2 ☒ 3 ☐ 4 ☐ 5

3. Los gráficos son agradables a la vista:

- ☐ 1 ☒ 2 ☐ 3 ☐ 4 ☐ 5

4. Crees que el tiempo es suficiente para llegar a la meta:

- ☐ 1 ☒ 2 ☐ 3 ☐ 4 ☐ 5

MOTIVACIÓN

1. Te ha hecho motivarte para conseguir llegar al final:

- ☐ 1 ☐ 2 ☐ 3 ☒ 4 ☐ 5

EMOCIÓN

1. Te han gustado los efectos sonoros:

- ☒ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

2. Te ha despertado emoción (tensión, alegría, frustración...) mientras jugabas:

- ☐ 1 ☐ 2 ☐ 3 ☒ 4 ☐ 5

ANÁLISIS PERSONAL

1. ¿Qué es lo que más te ha gustado?

te incita a continuar jugando hasta superar la pantalla en la que te encuentras.

2. ¿Qué es lo que menos te ha gustado?

La explicación de las instrucciones para jugar

3. ¿Qué mejorarías?

de una pantalla a otra se pierde la perspectiva de la posición del jugador en la pantalla.

4. ¿Qué nombre le pondrías al juego?

boltri

5. Valoración general

7

Joaquín López Pérez, 35 años.

TEST DE EVALUACIÓN DEL VIDEOJUEGO

Evalúa de 1 (muy poco) a 5 (mucho) los siguientes aspectos del juego. Debes ser totalmente sincero, gracias a tus respuestas se hará un análisis eficiente de videojuego.

ANÁLISIS DE LA JUGABILIDAD

SATISFACCIÓN

1. Encuentras divertido el juego:
- ☐ 1 ☐ 2 ☒ 3 ☐ 4 ☐ 5

APRENDIZAJE

1. Has jugado alguna vez a un juego parecido:
- ☐ 1 ☒ 2 ☐ 3 ☐ 4 ☐ 5
2. Te ha sido difícil comprender la mecánica del juego:
- ☐ 1 ☐ 2 ☒ 3 ☐ 4 ☐ 5
3. Crees que la dificultad está bien ajustada a cada nivel:
- ☐ 1 ☐ 2 ☐ 3 ☒ 4 ☐ 5
4. Has sentido que has sido penalizado repetidamente por el mismo error:
- ☐ 1 ☐ 2 ☐ 3 ☒ 4 ☐ 5

EFFECTIVIDAD

1. Ha motivado que lo hayas completado hasta el final:
- ☒ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5
2. Lo has encontrado corto de duración:
- ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☒ 5

INMERSIÓN

1. Ha hecho que mantengas la atención durante todo el juego:
- ☐ 1 ☐ 2 ☐ 3 ☒ 4 ☐ 5

2. Estéticamente lo ves acertado:
- ☐ 1 ☐ 2 ☐ 3 ☒ 4 ☐ 5
3. Los gráficos son agradables a la vista:
- ☐ 1 ☐ 2 ☐ 3 ☒ 4 ☐ 5
4. Crees que el tiempo es suficiente para llegar a la meta:
- ☐ 1 ☐ 2 ☒ 3 ☐ 4 ☐ 5

MOTIVACIÓN

1. Te ha hecho motivarte para conseguir llegar al final:
- ☐ 1 ☒ 2 ☐ 3 ☐ 4 ☐ 5

EMOCIÓN

1. Te han gustado los efectos sonoros:
- ☐ 1 ☒ 2 ☐ 3 ☐ 4 ☐ 5
2. Te ha despertado emoción (tensión, alegría, frustración...) mientras jugabas:
- ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☒ 5

ANÁLISIS PERSONAL

1. ¿Qué es lo que más te ha gustado?

Los colores y los gráficos

2. ¿Qué es lo que menos te ha gustado?

Los efectos sonoros y que me no me ha motivado a llegar al final porque no sabía muy bien lo que tenía que hacer.

3. ¿Qué mejorarías?

Poner efectos sonoros más realistas.

4. ¿Qué nombre le pondrías al juego?

PELOTA-PUM

5. Valoración general
- 6

José María Manzanero Sanz, 36 años.

TEST DE EVALUACIÓN DEL VIDEOJUEGO

Evalúa de 1 (muy poco) a 5 (mucho) los siguientes aspectos del juego. Debes ser totalmente sincero, gracias a tus respuestas se hará un análisis eficiente de videojuego.

ANÁLISIS DE LA JUGABILIDAD

SATISFACCIÓN

1. Encuentras divertido el juego:

☐ 1 ☒ 2 ☐ 3 ☐ 4 ☐ 5

APRENDIZAJE

1. Has jugado alguna vez a un juego parecido:

☐ 1 ☒ 2 ☐ 3 ☐ 4 ☐ 5

2. Te ha sido difícil comprender la mecánica del juego:

☐ 1 ☐ 2 ☒ 3 ☐ 4 ☐ 5

3. Crees que la dificultad está bien ajustada a cada nivel:

☐ 1 ☐ 2 ☐ 3 ☒ 4 ☐ 5

4. Has sentido que has sido penalizado repetidamente por el mismo error:

☐ 1 ☐ 2 ☐ 3 ☒ 4 ☐ 5

EFFECTIVIDAD

1. Ha motivado que lo hayas completado hasta el final:

☐ 1 ☐ 2 ☐ 3 ☒ 4 ☐ 5

2. Lo has encontrado corto de duración:

☐ 1 ☐ 2 ☐ 3 ☒ 4 ☐ 5

INMERSIÓN

1. Ha hecho que mantengas la atención durante todo el juego:

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☒ 5

2. Estéticamente lo ves acertado:

☐ 1 ☐ 2 ☐ 3 ☒ 4 ☐ 5

3. Los gráficos son agradables a la vista:

☐ 1 ☐ 2 ☐ 3 ☒ 4 ☐ 5

4. Crees que el tiempo es suficiente para llegar a la meta:

☐ 1 ☐ 2 ☒ 3 ☐ 4 ☐ 5

MOTIVACIÓN

1. Te ha hecho motivarte para conseguir llegar al final:

☐ 1 ☐ 2 ☒ 3 ☐ 4 ☐ 5

EMOCIÓN

1. Te han gustado los efectos sonoros:

☐ 1 ☒ 2 ☐ 3 ☐ 4 ☐ 5

2. Te ha despertado emoción (tensión, alegría, frustración...) mientras jugabas:

☐ 1 ☐ 2 ☐ 3 ☒ 4 ☐ 5

ANÁLISIS PERSONAL

1. ¿Qué es lo que más te ha gustado?

Los gráficos

2. ¿Qué es lo que menos te ha gustado?

Me ha parecido muy corto

3. ¿Qué mejorarías?

Pondría más niveles

4. ¿Qué nombre le pondrías al juego?

BOLA-1

5. Valoración general

6

Julián Rodríguez Sánchez, 46 años.

TEST DE EVALUACIÓN DEL VIDEOJUEGO

Evalúa de 1 (muy poco) a 5 (mucho) los siguientes aspectos del juego. Debes ser totalmente sincero, gracias a tus respuestas se hará un análisis eficiente de videojuego.

ANÁLISIS DE LA JUGABILIDAD

SATISFACCIÓN

1. Encuentras divertido el juego:

☐ 1 ☐ 2 ☒ 3 ☐ 4 ☐ 5

APRENDIZAJE

1. Has jugado alguna vez a un juego parecido:

☒ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

2. Te ha sido difícil comprender la mecánica del juego:

☐ 1 ☐ 2 ☒ 3 ☐ 4 ☐ 5

3. Crees que la dificultad está bien ajustada a cada nivel:

☐ 1 ☐ 2 ☐ 3 ☒ 4 ☐ 5

4. Has sentido que has sido penalizado repetidamente por el mismo error:

☐ 1 ☐ 2 ☒ 3 ☐ 4 ☐ 5

EFFECTIVIDAD

1. Ha motivado que lo hayas completado hasta el final:

☐ 1 ☐ 2 ☒ 3 ☐ 4 ☐ 5

2. Lo has encontrado corto de duración:

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☒ 5

INMERSIÓN

1. Ha hecho que mantengas la atención durante todo el juego:

☐ 1 ☐ 2 ☒ 3 ☐ 4 ☐ 5

2. Estéticamente lo ves acertado:

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☒ 5

3. Los gráficos son agradables a la vista:

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☒ 5

4. Crees que el tiempo es suficiente para llegar a la meta:

☐ 1 ☐ 2 ☐ 3 ☒ 4 ☐ 5

MOTIVACIÓN

1. Te ha hecho motivarte para conseguir llegar al final:

☐ 1 ☐ 2 ☐ 3 ☒ 4 ☐ 5

EMOCIÓN

1. Te han gustado los efectos sonoros:

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☒ 5

2. Te ha despertado emoción (tensión, alegría, frustración...) mientras jugabas:

☐ 1 ☐ 2 ☒ 3 ☐ 4 ☐ 5

ANÁLISIS PERSONAL

1. ¿Qué es lo que más te ha gustado?

los gráficos y la decoración

2. ¿Qué es lo que menos te ha gustado?

Los controles

3. ¿Qué mejorarías?

El control sobre la bola

4. ¿Qué nombre le pondrías al juego?

BOLA2011

5. Valoración general

8

Paloma Torres Puerta, 36 años.

TEST DE EVALUACIÓN DEL VIDEOJUEGO

Evalúa de 1 (muy poco) a 5 (mucho) los siguientes aspectos del juego. Debes ser totalmente sincero, gracias a tus respuestas se hará un análisis eficiente de videojuego.

ANÁLISIS DE LA JUGABILIDAD

SATISFACCIÓN

1. Encuentras divertido el juego:

☐ 1 ☐ 2 ☐ 3 ☒ 4 ☐ 5

APRENDIZAJE

1. Has jugado alguna vez a un juego parecido:

☐ 1 ☒ 2 ☐ 3 ☐ 4 ☐ 5

2. Te ha sido difícil comprender la mecánica del juego:

☐ 1 ☒ 2 ☐ 3 ☐ 4 ☐ 5

3. Crees que la dificultad está bien ajustada a cada nivel:

☐ 1 ☐ 2 ☐ 3 ☒ 4 ☐ 5

4. Has sentido que has sido penalizado repetidamente por el mismo error:

☐ 1 ☐ 2 ☐ 3 ☒ 4 ☐ 5

EFFECTIVIDAD

1. Ha motivado que lo hayas completado hasta el final:

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☒ 5

2. Lo has encontrado corto de duración:

☐ 1 ☒ 2 ☐ 3 ☐ 4 ☐ 5

INMERSIÓN

1. Ha hecho que mantengas la atención durante todo el juego:

☐ 1 ☐ 2 ☒ 3 ☐ 4 ☐ 5

2. Estéticamente lo ves acertado:

☐ 1 ☐ 2 ☒ 3 ☐ 4 ☐ 5

3. Los gráficos son agradables a la vista:

☐ 1 ☐ 2 ☒ 3 ☐ 4 ☐ 5

4. Crees que el tiempo es suficiente para llegar a la meta:

☐ 1 ☐ 2 ☐ 3 ☒ 4 ☐ 5

MOTIVACIÓN

1. Te ha hecho motivarte para conseguir llegar al final:

☒ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

EMOCIÓN

1. Te han gustado los efectos sonoros:

☐ 1 ☐ 2 ☐ 3 ☒ 4 ☐ 5

2. Te ha despertado emoción (tensión, alegría, frustración...) mientras jugabas:

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☒ 5

ANÁLISIS PERSONAL

1. ¿Qué es lo que más te ha gustado?

Los sonidos

2. ¿Qué es lo que menos te ha gustado?

Me he puesto un poco nerviosa y me ha costado mucho llegar al final

3. ¿Qué mejorarías?

Los colores

4. ¿Qué nombre le pondrías al juego?

PELOTA

5. Valoración general

6

Rafael Román Medina, 53 años.

TEST DE EVALUACIÓN DEL VIDEOJUEGO

Evalúa de 1 (muy poco) a 5 (mucho) los siguientes aspectos del juego. Debes ser totalmente sincero, gracias a tus respuestas se hará un análisis eficiente de videojuego.

ANÁLISIS DE LA JUGABILIDAD

SATISFACCIÓN

1. Encuentras divertido el juego:

☐ 1 ☐ 2 ☐ 3 ☒ 4 ☐ 5

APRENDIZAJE

1. Has jugado alguna vez a un juego parecido:

☒ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

2. Te ha sido difícil comprender la mecánica del juego:

☐ 1 ☒ 2 ☐ 3 ☐ 4 ☐ 5

3. Crees que la dificultad está bien ajustada a cada nivel:

☐ 1 ☐ 2 ☐ 3 ☒ 4 ☐ 5

4. Has sentido que has sido penalizado repetidamente por el mismo error:

☐ 1 ☐ 2 ☐ 3 ☒ 4 ☐ 5

EFFECTIVIDAD

1. Ha motivado que lo hayas completado hasta el final:

☐ 1 ☐ 2 ☐ 3 ☒ 4 ☐ 5

2. Lo has encontrado corto de duración:

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☒ 5

INMERSIÓN

1. Ha hecho que mantengas la atención durante todo el juego:

☐ 1 ☐ 2 ☐ 3 ☒ 4 ☐ 5

2. Estéticamente lo ves acertado:

☐ 1 ☐ 2 ☐ 3 ☒ 4 ☐ 5

3. Los gráficos son agradables a la vista:

☐ 1 ☐ 2 ☐ 3 ☒ 4 ☐ 5

4. Crees que el tiempo es suficiente para llegar a la meta:

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☒ 5

MOTIVACIÓN

1. Te ha hecho motivarte para conseguir llegar al final:

☐ 1 ☐ 2 ☒ 3 ☐ 4 ☐ 5

EMOCIÓN

1. Te han gustado los efectos sonoros:

☐ 1 ☐ 2 ☒ 3 ☐ 4 ☐ 5

2. Te ha despertado emoción (tensión, alegría, frustración...) mientras jugabas:

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☒ 5

ANÁLISIS PERSONAL

1. ¿Qué es lo que más te ha gustado?

Los colores de los niveles

2. ¿Qué es lo que menos te ha gustado?

No está bien explicado lo que hay que hacer

3. ¿Qué mejorarías?

La ayuda

4. ¿Qué nombre le pondrías al juego?

Pin-Pon

5. Valoración general

6

Juan Carlos Rodríguez Sánchez, 40 años.

TEST DE EVALUACIÓN DEL VIDEOJUEGO

Evalúa de 1 (muy poco) a 5 (mucho) los siguientes aspectos del juego. Debes ser totalmente sincero, gracias a tus respuestas se hará un análisis eficiente de videojuego.

ANÁLISIS DE LA JUGABILIDAD

SATISFACCIÓN

1. Encuentras divertido el juego:

APRENDIZAJE

1. Has jugado alguna vez a un juego parecido:

2. Te ha sido difícil comprender la mecánica del juego:

3. Crees que la dificultad está bien ajustada a cada nivel:

4. Has sentido que has sido penalizado repetidamente por el mismo error:

EFFECTIVIDAD

1. Ha motivado que lo hayas completado hasta el final:

2. Lo has encontrado corto de duración:

INMERSIÓN

1. Ha hecho que mantengas la atención durante todo el juego:

2. Estéticamente lo ves acertado:

3. Los gráficos son agradables a la vista:

4. Crees que el tiempo es suficiente para llegar a la meta:

MOTIVACIÓN

1. Te ha hecho motivarte para conseguir llegar al final:

EMOCIÓN

1. Te han gustado los efectos sonoros:

2. Te ha despertado emoción (tensión, alegría, frustración...) mientras jugabas:

ANÁLISIS PERSONAL

1. ¿Qué es lo que más te ha gustado?

movimiento de la plataforma

2. ¿Qué es lo que menos te ha gustado?

fondo de pantalla

3. ¿Qué mejorarías?

mejores vistas espaciales para aumentar la dificultad de movimientos

4. ¿Qué nombre le pondrías al juego?

ball.space

5. Valoración general

4

189